KUKA

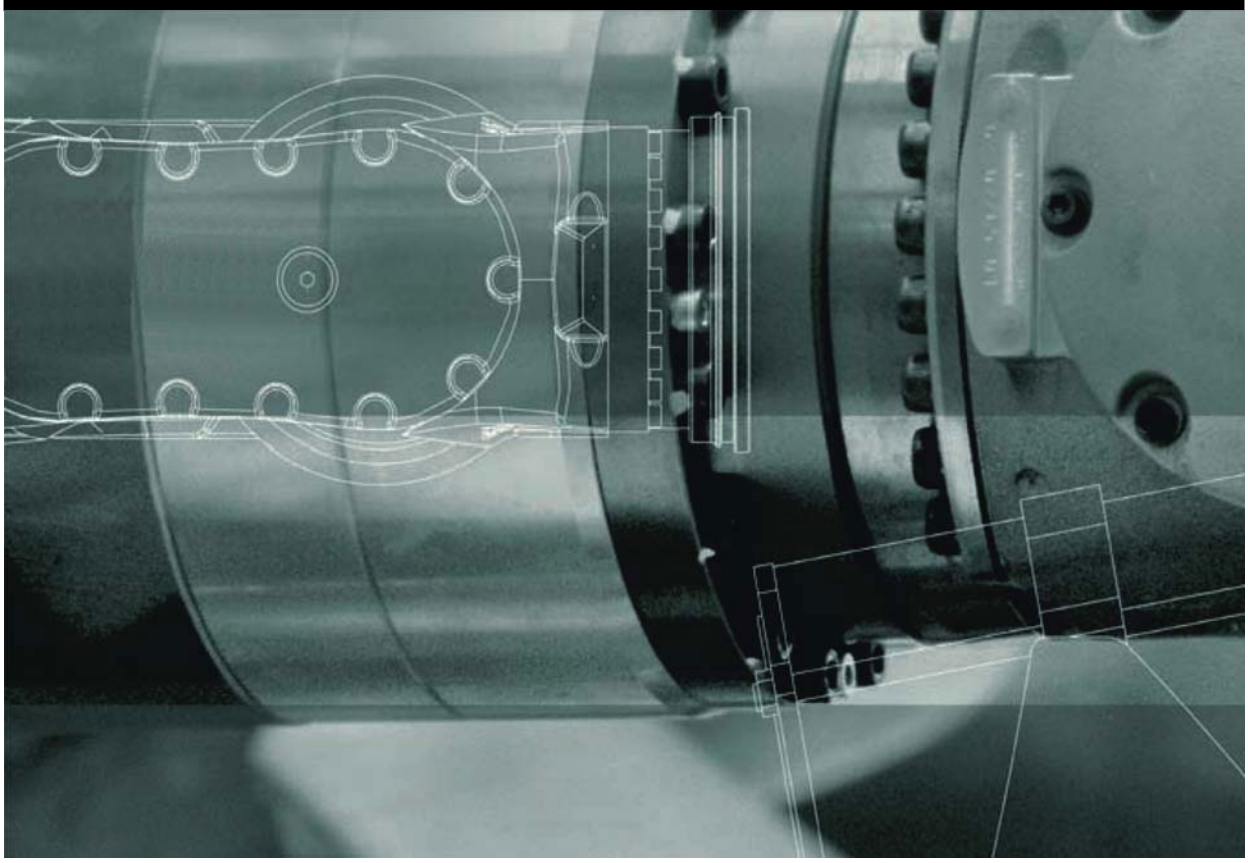
Controller Option

KUKA Roboter GmbH

Interbus 1.1

For KUKA System Software 8.2

For VW System Software 8.2



Issued: 29.11.2012

Version: KR C4 Interbus 1.1 V2 en (PDF)

KUKA Roboter GmbH
Zugspitzstraße 140
D-86165 Augsburg
Germany

Other functions not described in this documentation may be operable in the controller. The user has no claims to these functions, however, in the case of a replacement or service work.

We have checked the content of this documentation for conformity with the hardware and software described. Nevertheless, discrepancies cannot be precluded, for which reason we are not able to guarantee total conformity. The information in this documentation is checked on a regular basis, however, and necessary corrections will be incorporated in the subsequent edition.

Subject to technical alterations without an effect on the function.

Translation of the original documentation

KIM-PS5-DOC

Publication:	Pub KR C4 Interbus 1.1 (PDF) en
Bookstructure:	KR C4 Interbus 1.1 V2.1
Version:	KR C4 Interbus 1.1 V2 en (PDF)

Contents

1 Introduction

1.1 Target group

This documentation is aimed at users with the following knowledge and skills:

- Advanced KRL programming skills
- Advanced knowledge of the robot controller system
- Advanced knowledge of bus systems

For optimal use of our products, we recommend that our customers take part in a course of training at KUKA College. Information about the training program can be found at www.kuka.com or can be obtained directly from our subsidiaries.

1.2 Industrial robot documentation

The industrial robot documentation consists of the following parts:

- Documentation for the manipulator
- Documentation for the robot controller
- Operating and programming instructions for the KUKA System Software
- Documentation relating to options and accessories
- Parts catalog on storage medium

Each of these sets of instructions is a separate document.

1.3 Representation of warnings and notes

Safety

These warnings are relevant to safety and **must** be observed.

DANGER These warnings mean that it is certain or highly probable that death or severe injuries **will** occur, if no precautions are taken.

WARNING These warnings mean that death or severe injuries **may** occur, if no precautions are taken.

CAUTION These warnings mean that minor injuries **may** occur, if no precautions are taken.

NOTICE These warnings mean that damage to property **may** occur, if no precautions are taken.

These warnings contain references to safety-relevant information or general safety measures.
These warnings do not refer to individual hazards or individual precautionary measures.

This warning draws attention to procedures which serve to prevent or remedy emergencies or malfunctions:

SAFETY INSTRUCTIONS Procedures marked with this warning **must** be followed exactly.

Notes

These hints serve to make your work easier or contain references to further information.

Tip to make your work easier or reference to further information.

1.4 Trademarks

Windows is a trademark of Microsoft Corporation.

1.5 Terms used

Term	Description
CMD	Configuration, Monitoring, Diagnostic: software for configuration, monitoring and troubleshooting in Interbus interfaces.
CR	Communication reference.
DTM	Device Type Manager Device description file
I/O	Inputs/outputs, e.g. at terminals and on machines.
Remote bus	Designation for the main ring of an Interbus system
FSMA	Field-installable SubMiniature Assembly: Fiber-optic connector with screw lock, outwardly similar to the electrical SMA connector.
FW	Firmware: normally unchangeable operating software of a device which is automatically loaded when the device is activated.
HCS fiber	HCS fiber (Hard Cladded Silica): a FOC variant consisting of a glass fiber core with a plastic cladding.
HW	Hardware: physical, electronic components and modules.
IBS	Interbus
Interbus	A field bus introduced by Phoenix Contact. It is defined as a ring system in which every device regenerates and forwards the incoming signal. The Interbus offers high data throughput at a low cycle rate and is particularly immune to interference. It is defined for normal copper cables and also for fiber-optic cables. Up to 512 slaves can be connected to an Interbus (master-slave structure) and up to 4096 I/Os can be served. The main ring (remote bus) can contain up to 256 devices; local buses or loop segments can be coupled, e.g. in a machine, by means of bus couplers.
ISA	Industry Standard Architecture: commonly-used PC bus before PCI.

Term	Description
KCP	The KCP (KUKA Control Panel) teach pendant has all the operator control and display functions required for operating and programming the industrial robot. The KCP variant for the KR C4 is called KUKA smartPAD. The general term “KCP”, however, is generally used in this documentation.
Configuration file	Text files with specifications for parameters and settings.
KR C	KUKA Robot Controller.
KRL	KUKA Robot Language: KUKA robot programming language.
KUKA.HMI	The KUKA user interface on the screen of the robot controller.
Local bus	Designation for the sub-rings of an Interbus system.
FOC	Fiber-optic cables: made of glass or plastic fibers. Greater immunity to interference than copper cables; electrical potential differences have no effect.
MAU	Medium Attachment Unit: connection unit for a bus device.
MAU warning	Warning of decreasing transmission quality/ weakening reception level on the optic transmission link, in order to be able to isolate and eliminate the cause (maladjustment, dirt, etc.) before it leads to a malfunction.
MPM	Multi-Port Memory: Interbus component that communicates between the bus and the processors.
PCI	Peripheral Component Interconnect: PC bus for coupling plug-in cards to the processor.
PCP	Peripherals Communication Protocol: protocol for sending info telegrams (e.g. message texts to be displayed) via an Interbus. It is not actively used by robot controllers, but forwarded.
PDU	Protocol Data Units: data packets on the Interbus
Polymer fiber cable	Plastic cable. Cheaper than glass fiber or HCS fiber, but with a shorter range.
PLC	Programmable Logic Controller: is used in systems as a higher-level master module in the bus system.
SW	Software
Telnet	Simple terminal communications protocol. It is used, for example, to configure individual hardware units.

2 Product description

2.1 Overview

The Interbus PCI interface is a combined master/slave controller board. Master and slave are housed on two separate PCI plug-in cards and equipped with their own processors.

The factory-installed firmware can be updated via the serial diagnostic interface. The parameterization can be saved in a non-volatile memory on the module. Process data channels and parameter channels (PCP) are supported.

Configuration, mapping and diagnosis can be carried out using the Config+ tool from Phoenix Contact or WorkVisual from KUKA Roboter GmbH.

> The range of functions of the G4 firmware (= 4th generation) is restricted by the driver as follows: the driver does not support the distribution of process data from the slave to multiple address blocks. The slave process data range must thus be located in a single block starting at a freely selected address in the MPM.

The Interbus enables communication between the robot controller and the various I/O units.

The Interbus PCI controller board is available in two versions: one for connection via copper cables and one for fiber-optic cables.

Interbus is a field bus system that works with a ring structure and active coupling between the devices. The bus access procedure is a master/slave system. The data are passed from the master to the slaves as if through a shift register.

Properties

- Data transmission via ring system
- Master/slave system
- Connected devices are automatically loaded by means of an identification cycle
- Supports PCP functionality
- Configurable and parameterizable across the network
- No terminating resistors
- No device address settings have to be made on-site
- Can be expanded simply
- Flexible adaptation to system topology
- Diagnostic options
- Transmission speed of 500 or 2000* kBaud (baud rate). Transmission at 2000 kBaud must be supported by all devices.
- I/O units can be switched on/off
- Branches are easy to implement using remote bus branch terminals
- Branches can be cascaded freely
- Maximum of 512 devices, of which max. 254 remote bus devices
- Maximum of 62 PCP devices
- Maximum cable length between the slave devices: 400 m
- Maximum cable length with copper cables: 13 km
- Maximum cable length with polymer fibers: 70 m
- Maximum cable length with HCS (hard-clad silica): 400 m
- Maximum cable length with fiberglass cable: 3500 m
- Optional operation of master and slave at 500 kBaud or 2 MBaud

- Up to 4096 I/Os (FW version 4.49)
- Up to 64 PCP devices (FW version 4.49)
- The slave part is supplied independently via an external 24 V power supply

Compatibility

KR C4 Interbus 1.1 is compatible with the following field buses:

- KR C4 PROFINET 2.0, 2.1 and 2.2
- KR C4 EtherCAT

2.2 IBS PCI SC controller board

The IBS PCI SC controller board, the Interbus PCI interface for the KR C4 robot controller, is available in two versions:

- IBS PCI SC/RI-LK for connection of fiber-optic cables
- IBS PCI SC/RI/I-T for connection of copper cables

(>>> 4.1.1 "PCI slot assignment" Page 13)

The controller board consists either of both master and slave cards or just a master card. The master card can also be installed and operated without a slave part. The slave card, however, can only be installed and operated in combination with a master card.

The master card is installed in PCI slot 1 and the slave card is installed in PCI slot 2.

2.3 PCP functionality

The slave part of the Interbus PCI interface supports the Peripherals Communication Protocol. PCP makes it possible to access the lower-level Interbus from a higher-level Interbus via the MPM (I/O range). By reading from or writing to the MPM from the higher-level Interbus, it is possible to read inputs and outputs in the lower-level Interbus and to set outputs there.

No PCP data are sent to the robot controller. The PCP functionality is restricted to the Interbus driver and to reading and writing I/O data in the MPM from the higher-level controller. When making the parameter channel and process data channel settings for the PCP functionality, the DIP switch settings on the slave module (>>> 6.3 "DIP switches on the slave module" Page 22) must be observed.

3 Safety

This documentation contains safety instructions which refer specifically to the product described here. The fundamental safety information for the industrial robot can be found in the “Safety” chapter of the operating or assembly instructions for the robot controller.

⚠ WARNING The “Safety” chapter in the operating instructions or assembly instructions of the robot controller must be observed. Death to persons, severe injuries or considerable damage to property may otherwise result.

4 Installation

4.1 System requirements

Robot controller

Hardware:

- KR C4

Software:

- KUKA System Software 8.2
- Or VW System Software 8.2

Laptop/PC

- The system requirements for installation of WorkVisual are contained in the WorkVisual documentation.

4.1.1 PCI slot assignment

Description

The IBS PCI SC controller board, the Interbus PCI interface for the KR C4 robot controller, is installed as follows:

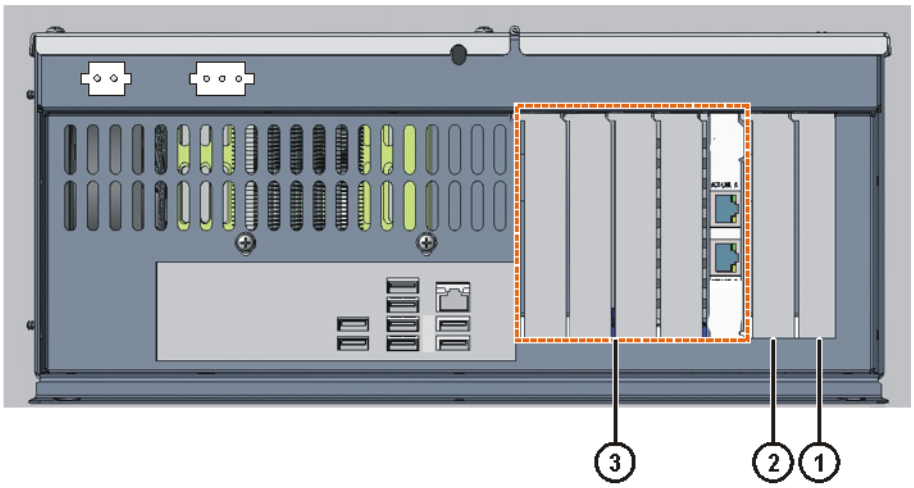


Fig. 4-1: Installing the master and slave cards

Slot	Designation	Plug-in card
1	PCI1	Interbus master card
2	PCI2	Interbus slave card
3 to 7	-	Field bus, slots 3 to 7

The controller board consists either of both master and slave cards or just a master card. The master card can also be installed and operated without a slave part. The slave card, however, can only be installed and operated in combination with a master card.

> The KR C4 cannot be operated with multiple master or slave cards.

4.2 Installing or updating INTERBUS (KSS)

Description

There is an option CD for INTERBUS:

- KR C4 Interbus 1.0

If a version of INTERBUS is already installed, its configuration is carried over automatically. If this is not desired, the existing version must first be uninstalled.

> It is advisable to archive all relevant data before updating a software package.

Precondition

- Software on KUKA.USBData stick
- No program is selected.
- T1 or T2 operating mode
- “Expert” user group

> **NOTICE** Only the KUKA.USB data stick may be used. Data may be lost or modified if any other USB stick is used.

Procedure

1. Plug in USB stick.
2. Select **Start-up** > **Install additional software** in the main menu.
3. Press **New software**. If a software package that is on the USB stick is not displayed, press **Refresh**.
4. Select the entry **KRC4 Interbus** and press **Install**. Reply to the request for confirmation with **Yes**. The files are copied onto the hard drive.
5. Repeat step 4 if another software package is to be installed from this stick.
6. Remove USB stick.
7. It may be necessary to reboot the controller, depending on the additional software. In this case, a corresponding prompt is displayed. Confirm with **OK** and reboot the robot controller. Installation is resumed and completed.

LOG file

A LOG file is created under C:\KRC\ROBOTER\LOG.

4.3 Installing INTERBUS (VSS)

The KR C4 Interbus 1.0 option CD is included in VSS 8.2. KR C4 Interbus 1.0 is automatically installed together with VSS 8.2.

4.4 Uninstalling INTERBUS (KSS)

> It is advisable to archive all relevant data before uninstalling a software package.

Precondition

- “Expert” user group

Procedure

1. In the main menu, select **Start-up** > **Additional software**. All additional programs installed are displayed.
2. Select the entry **KRC4 Interbus** and press **Uninstall**. Reply to the request for confirmation with **Yes**. Uninstallation is prepared.
3. Reboot the robot controller. Uninstallation is resumed and completed.

LOG file

A LOG file is created under C:\KRC\ROBOTER\LOG.

5 Start-up and recommissioning

5.1 IBS PCI SC/RI-LK

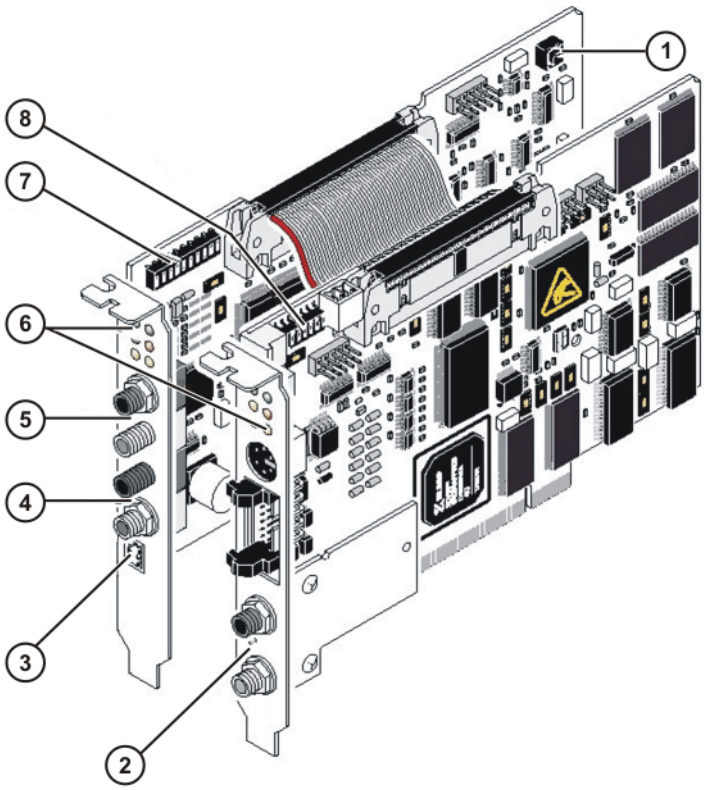


Fig. 5-1: Configuration of the IBS PCI SC/RI-LK controller board

1 Reset button (slave)
2 Master interface (Remote Out, outgoing remote bus)
3 External 24 V supply voltage (slave)
4 Slave interface (Remote Out, outgoing remote bus)
5 Slave interface (Remote In, incoming remote bus)
6 Indicator elements (LEDs)
7 DIP switches for the slave configuration
8 DIP switches for the master configuration

5.2 Connecting fiber-optic cables

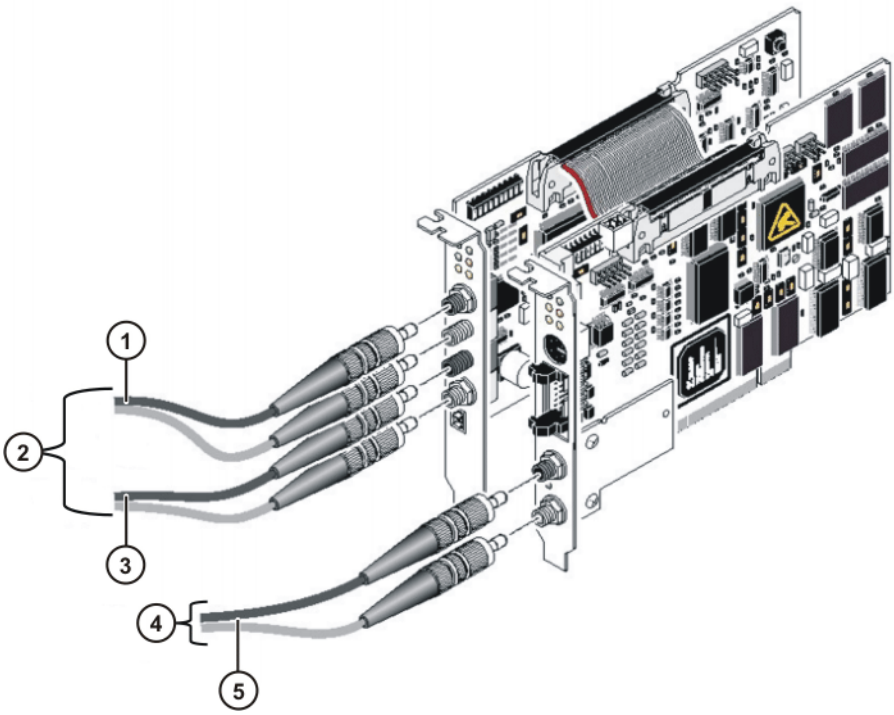


Fig. 5-2: Connecting the fiber-optic cables

1 Remote IN, incoming remote bus
2 Slave
3 Remote OUT, outgoing remote bus of the slave card
4 Master
5 Remote OUT, outgoing remote bus of the master card

The IBS PCI SC/RI-LK can work with HCS and polymer fiber cables with FSMA connectors. The connectors must be secured with union nuts.

5.3 CMD interface

The CMD configuration and diagnostic software or Config+ from Phoenix Contact can access the IBS PCI SC controller board via the RS232 serial interface. Using CMD, the user can configure, parameterize and diagnose the Interbus. The parameterization and configuration can be stored in a non-volatile memory on the controller board using CMD. It is also possible to update the firmware of the IBS PCI SC controller board via the RS232 interface.

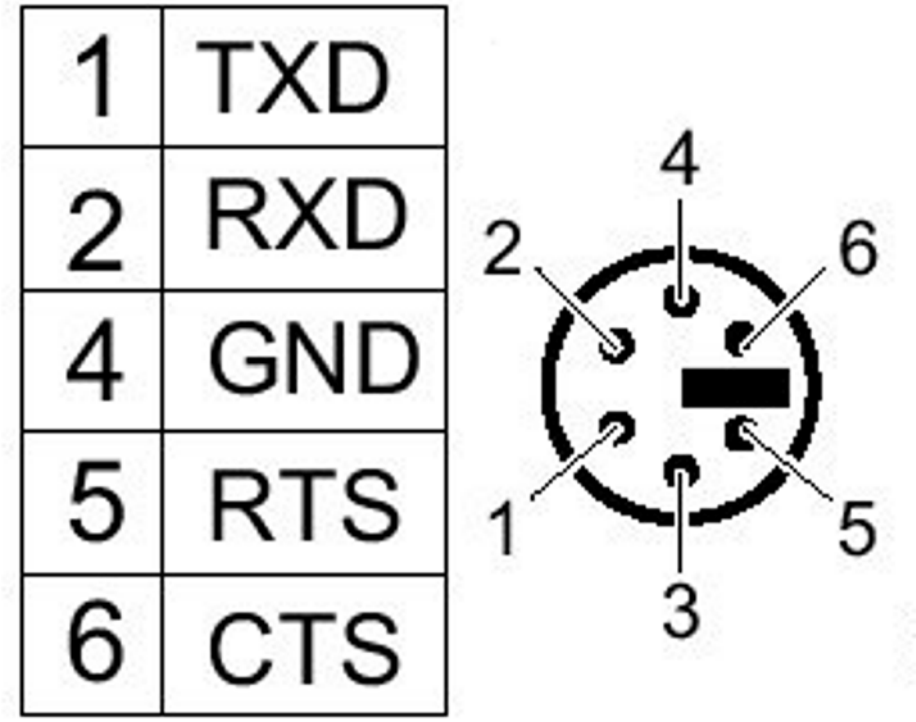


Fig. 5-3: CMD interface: 6-pin mini-DIN socket (PS/2)

The CMD interface is designed as a 6-contact mini-DIN socket (PS/2) on the front plate.

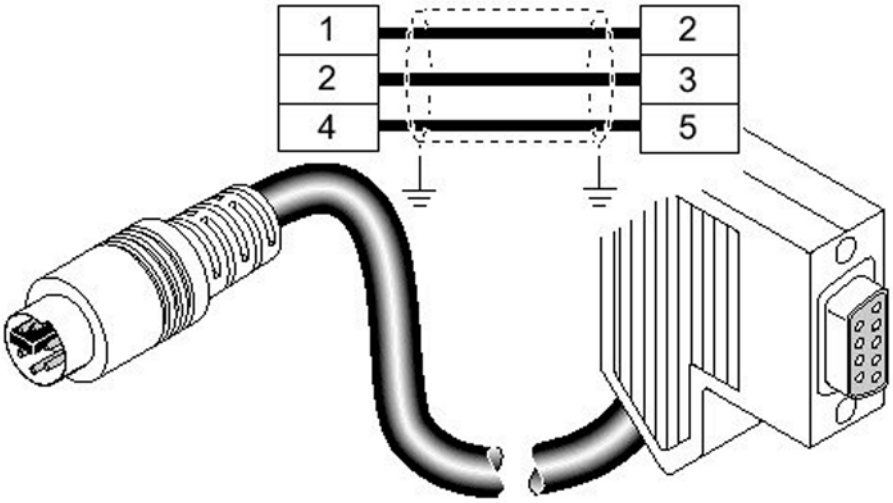


Fig. 5-4: RS232 cable for connection to diagnostic PC

It is connected to the diagnostic PC via a special RS232 cable that is available from Phoenix Contact.

5.4 IBS PCI SC/RI/I-T

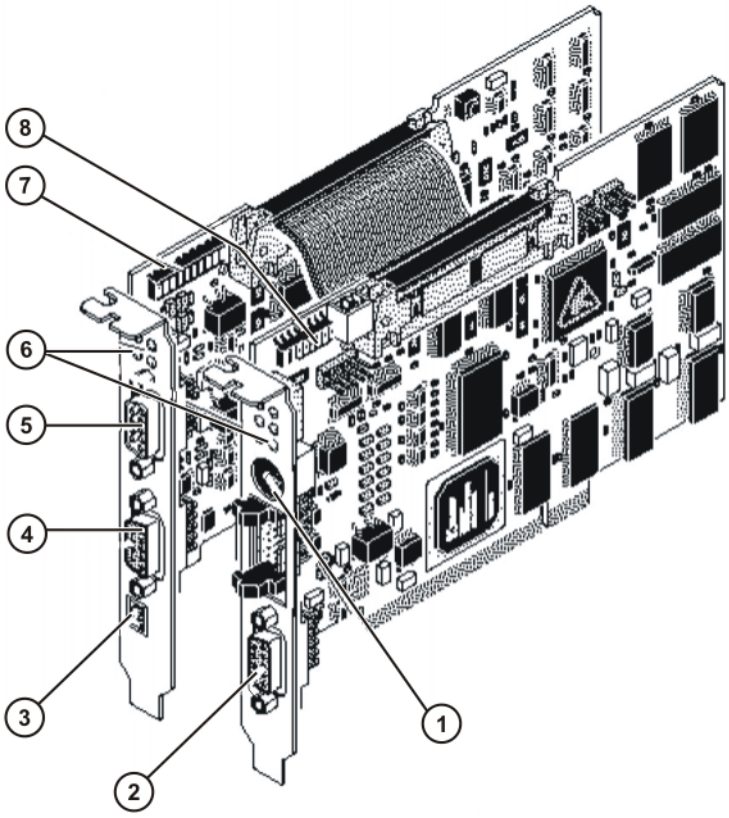


Fig. 5-5: Configuration of the IBS PCI SC/RI/I-T controller board

1 RS232 interface (CMD connection)
2 Master interface (Remote Out, outgoing remote bus)
3 External 24 V supply voltage (slave)
4 Slave interface (Remote Out, outgoing remote bus)
5 Slave interface (Remote In, incoming remote bus)
6 Indicator elements (LEDs)
7 DIP switches for the slave configuration
8 DIP switches for the master configuration

5.5 External power supply to slave

An external 24 V DC power supply is required for operation of the slave module. This is connected via a 2-pin MINI-COMBICON connector.

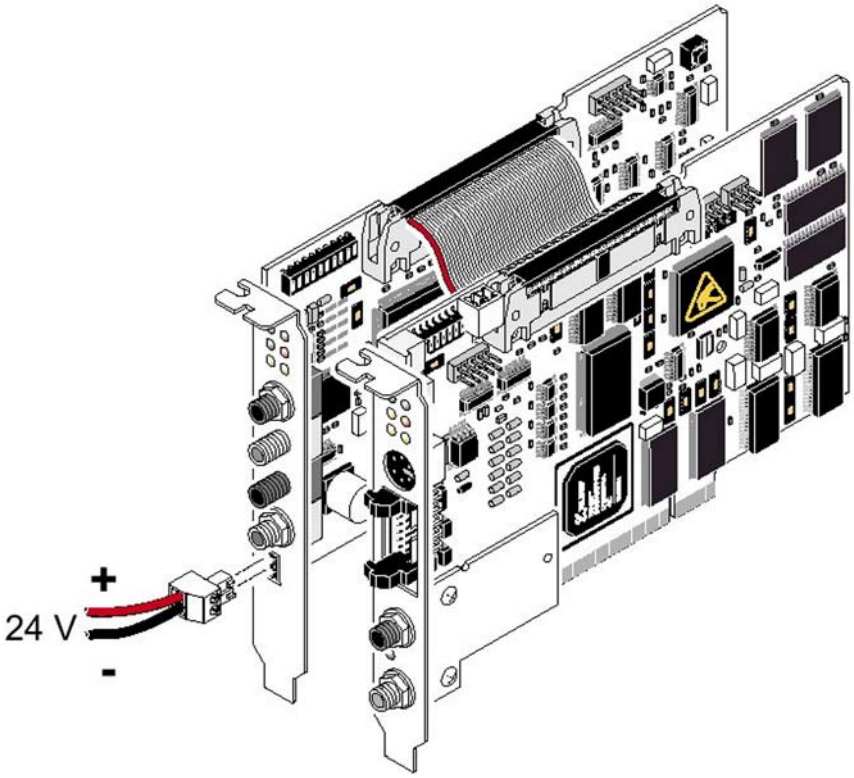


Fig. 5-6: Connection of the external 24 V power supply to the slave module

6 Configuration

6.1 Overview

Step	Description
1	DIP switches of master module (>>> 6.2 "DIP switches on the master module" Page 21)
2	Only if INTERBUS slave present. DIP switches of slave module (>>> 6.3 "DIP switches on the slave module" Page 22)
3	Configure the configuration file IBSPCI.XML. (>>> 6.4 "Configuring the file IBSPCI.XML" Page 24)
4	Configure INTERBUS with WorkVisual. (>>> 6.5 "Configuring the bus with WorkVisual" Page 28)
5	Map the inputs and outputs in WorkVisual.
6	Transfer the bus configuration from WorkVisual to the robot controller.

Additional information about procedures in WorkVisual is contained in the WorkVisual documentation.

6.2 DIP switches on the master module

The DIP switches are on the top left-hand side of the master module.

KUKA default: DIP 1 … 3 OFF

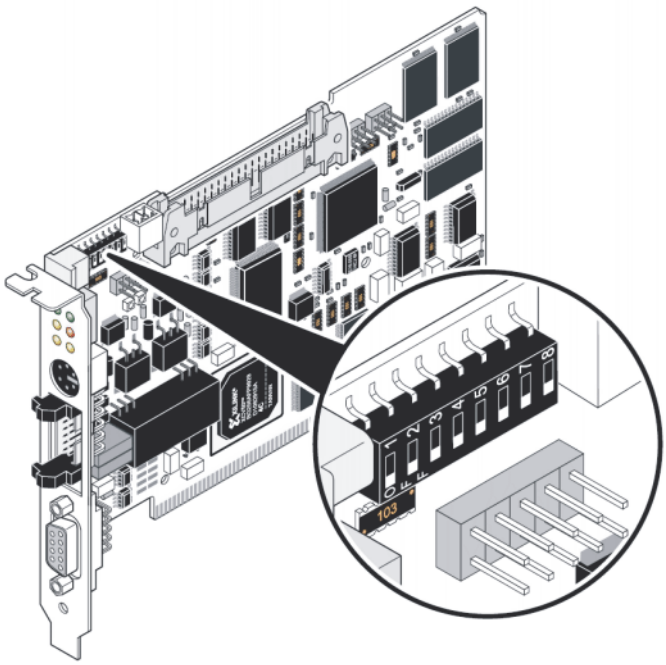


Fig. 6-1: DIP switches on the master module

1 ... 3: Card number

DIP switches 1 to 3 are used for setting the card number. If multiple Interbus cards are used, a card number must be assigned to each one. This number is used to identify the different cards in the system. The card number can be set

to any value between 1 and 8. The default value is 1. It is not necessary to change the card number if only one Interbus card is used.

> The card number must be specified when the driver is installed. It is advisable to make a note of it after making the setting.

Card number	DIP switch 1	DIP switch 2	DIP switch 3
1 (default)	OFF	OFF	OFF
2	ON	OFF	OFF
3	OFF	ON	OFF
4	ON	ON	OFF
5	OFF	OFF	ON
6	ON	OFF	ON
7	OFF	ON	ON
8	ON	ON	ON

4 ... 6: Additions

DIP switches 4 to 6 are reserved for expansions and must not be changed.

7: Baud rate

DIP switch 7 is used to set the baud rate. The DIP switch is set to OFF by default, i.e. the baud rate is detected automatically. This setting must not be changed.

8: Test mode

DIP switch 8 is used to activate the test mode. If the system is rebooted with test mode active, the controller board starts up the Interbus with physical addressing and activates it. During test mode, the controller board does not respond to instructions from the host system (PC). The controller board initializes the Interbus system and then starts it up automatically. Outputs are not set.

> In normal operation of the controller board, test mode must be deactivated by setting switch 8 to OFF.

6.3 DIP switches on the slave module

The DIP switches are on the top left-hand side of the slave module.

KUKA default: DIP 1 … 4 OFF, DIP 5 ON, DIP 6 … 9 OFF, DIP 10 ON

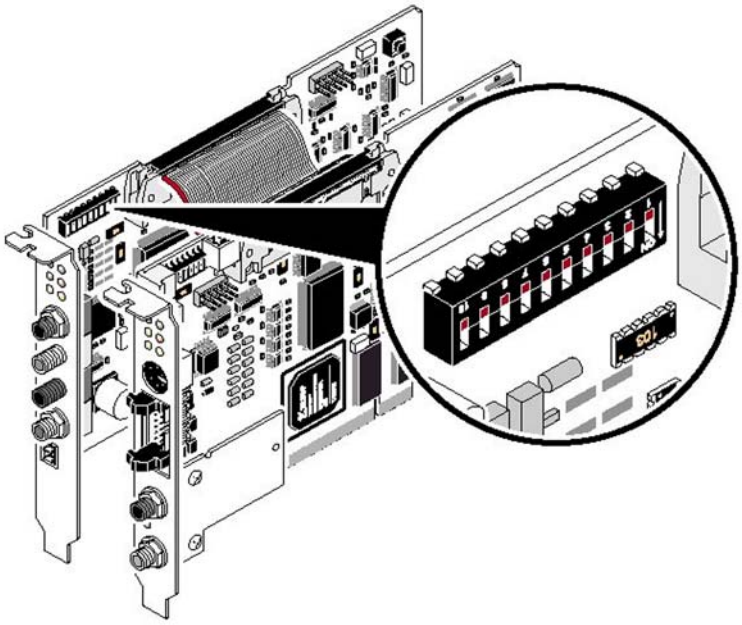


Fig. 6-2: DIP switches on the slave module

1, 2: Parameter channel

DIP switches 1 and 2 are used for setting the parameter channel (PCP). This setting also defines the ID code of the remote interface. The parameter channel and the process data channel can have a maximum width of 16 words.

DIP 1	DIP 2	Parameter channel	ID code (decimal)
OFF	OFF	0 words	3
ON	OFF	1 word	235
OFF	ON	2 words	232
ON	ON	4 words	233

3 ... 6: Process data length

DIP switches 3 to 6 are used for setting the process data length. The length of the process data also defines the length code.

DIP 3	DIP 4	DIP 5	DIP 6	Process data	Length code (decimal)
OFF	OFF	OFF	OFF	0 words	0
ON	OFF	OFF	OFF	1 word	1
OFF	ON	OFF	OFF	2 words	2
ON	ON	OFF	OFF	3 words	3
OFF	OFF	ON	OFF	4 words	4
ON	OFF	ON	OFF	5 words	5
OFF	ON	ON	OFF	6 words	6
ON	ON	ON	OFF	7 words	7
OFF	OFF	OFF	ON	8 words	8
ON	OFF	OFF	ON	9 words	9
OFF	ON	OFF	ON	10 words	10
ON	ON	OFF	ON	11 words	11
OFF	OFF	ON	ON	12 words	12
ON	OFF	ON	ON	13 words	13
OFF	ON	ON	ON	14 words	14
ON	ON	ON	ON	16 words	16

The module can be adapted to special requirements by setting the width of the parameter channel and the process data length. The following combinations are possible:

Parameter channel	Process data length (in words)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	16
0 words		X	X	X	X	X	X	X	X	X	X		X		X	X
1 word	X	X	X	X	X	X	X	X	X	X		X		X		
2 words	X	X	X	X	X	X	X		X		X		X		X	
4 words	X	X	X	X	X	X	X		X		X		X			

7: Reset response

DIP switch 7 determines whether a reset of the lower-level master system triggers a periphery fault in the higher-level system so that it can respond:

- OFF: no fault signaled in the higher-level system
- ON: fault signaled in the higher-level system

8: Reconfigure request

DIP switch 8 determines whether a Reconfigure request can be triggered via the OPC bus terminal:

- OFF: no Reconfigure request possible via the OPC bus terminal
- ON: Reconfigure request possible via the OPC bus terminal

9: Baud rate

DIP switch 9 determines the baud rate of the slave part of the controller board:

- OFF: 500 kbaud
- ON: 2 Mbaud

10: Configuration selection

DIP switch 10 determines whether DIP switches 1 to 9 are activated:

- OFF: DIP switches 1 to 9 ineffective; parameterization from stored resident configuration or from configuration received by lower-level master
- ON: DIP switches 1 to 9 determine parameterization

6.4 Configuring the file IBSPCI.XML

Description

The configuration file IBSPCI.XML contains all the settings which affect the Interbus connection.

Precondition

- Windows interface

Procedure

1. Open the file C:\KRC\ROBOTER\Config\User\Common\IBSPCI.XML.
2. Check the settings stored in the individual segments and adapt them if required.
3. Save the changes and close the file.

> Changes must always be made via the main menu **Configuration** > **Inputs/outputs** > **I/O drivers**.

The file IBSPCI.XML is divided into different sections.

<INTERBUS> section:

Entry	Value	Function
BOARDNUMBER	1 ... 8 Default: 1	Assigns a unique identifier to the controller board for the data channel. This identifier must match the card number selected by means of the DIP switches on the master module.

<CMD_CONFIGURATION> section:

Entry	Value	Function
CMD_FILE	-	Name of the SVC file. The path is predefined: KRC\Roboter\Config\User\Common directory.
MSGDELAY	Default: 10	Length of time in milliseconds that the system waits for a confirmation message after transmitting messages. If this time elapses, the message is considered not to have been confirmed.
USEBLOCKID	False True Default: True	This flag can be used to deactivate the use of the Block ID and the Block Offset. It should not normally be changed. False: Block ID and Block Offset are not evaluated. True: Block ID and Block Offset are evaluated.
BRK_IB_ERR	False True Default: True	If this flag is set, the loading of the SVC configuration file is aborted in the absence of a confirmation. False: Loading of the SVC configuration file is not aborted in the absence of a confirmation. True: Loading of the SVC configuration file is aborted in the absence of a confirmation.
TIMEOUT	50 ... 65000 Default: 60000	Maximum length of time in milliseconds used for various tasks, e.g. transmitting and receiving messages, during loading of the SVC configuration file.
BAUDRATE	Default: 9600	Baud rate of the CMD serial interface.
EXTERN_START	False True Default: False	False: Interbus is configured and started by the driver. True: Interbus is configured and started by an external tool, such as CMD, or a boot project. CMD_FILE entry is ignored.
COM_DLL_PORT	Default: 15001	Port number for communication between the configuration tools and the Interbus driver.

<CONFIGURATION> section:

Entry	Value	Function
SWAP_MASTER_BYTES	False True Default: True	This entry can be used to specify that the bytes of the modules in the master are to be swapped automatically by the driver. The driver only carries out this swapping, however, in the case of digital modules; analog modules are not swapped. False: The bytes are not swapped. True: The bytes are swapped.
RESET	0 ... 1 Default: 0	In order to set the controller board to a defined state when the driver is run up, a reset can be carried out before it is initialized. 0: No reset is carried out when the driver is run up. 1: A reset is carried out when the driver is run up; the system waits for a maximum of 7 s for the controller board to become operational once again. If it is not possible to address the controller board within these 7 s, the loading of the driver is aborted.

Entry	Value	Function
ERR_AUTO_QUIT_PF	False True Default: False	False: No acknowledgement of periphery faults. True: Automatic acknowledgement of periphery faults.
WATCHDOG	0 ... 7 Default: 0	The watchdog on the card is triggered every time data are read. If the watchdog is not triggered within the time specified, the controller board stops and displays a corresponding error message on the KUKA.HMI. The specified value sets the watchdog monitoring times according to the following list: 7: 16.4 ms 6: 32.8 ms 5: 65.5 ms 4: 131.1 ms 3: 262.1 ms 2: 524.3 ms 1: 1048.6 ms 0: Watchdog deactivated
DUMPFILE	-	Specification of a file name activates the ibsPciDump function. The name can be specified along with its path relative to the KRC/Roboter directory.
CONTINUE_WITH _WARN	False True Default: True	False: In the case of a periphery fault, the application is notified that an error has occurred while reading and writing the I/Os (like in the case of a bus error). As soon as the periphery fault has been rectified, this information is retracted. True: In the case of a periphery fault, the application is not notified that an error has occurred while reading and writing the I/Os.
MASTER_USED	False True Default: True	False: The master of the controller board is not initialized. True: The master of the controller board is initialized and started.
ERR_CLEAR_MPM_OUT	False True Default: False	This can prevent the outputs in the MPM from being set to 0 as soon as a bus error occurs in the master ring. False: Bus errors do not affect MPM outputs. True: Outputs in the MPM are set to 0 in the event of bus errors.

<DIAGNOSTIC_REGISTERS> section:

This section is not evaluated when the controller board is started via the SVC file or externally.

Entry	Value	Function
STATUS	Default: -1	I/O address for status register
PARAMETER1	Default: -1	I/O address for 1st parameter register
PARAMETER2	Default: -1	I/O address for 2nd parameter register

Entry	Value	Function
SLAVE_STATUS	Default: -1	I/O address for slave status register
SLAVE_PARAMETER	Default: -1	I/O address for slave parameter register

<TASK> section:

Entry	Value	Function
AUTORESTART	0 ... n Default: 0	0: Bus is not automatically restarted. 8 ... 20: An attempt is made, at the specified repeat rate in s, to restart the Interbus. n < 8 or n > 20: An attempt is made every 8 s to restart the Interbus.

<SLAVE> section:

Entry	Value	Function
SLAVE_ID	Default: 0x0403	The slave ID consists of two parts: the length of the slave process data in the High byte and the actual slave ID in the Low byte. Slave ID 3 thus designates a digital input and output module. If no slave ID has been specified, the slave is initialized with the ID 0x0403. This entry is not evaluated when the controller board is started via the SVC file or externally. If the slave ID has been changed, the slave module must be disconnected briefly from the external 24 V supply in order for the new data to be accepted.
SWAP_SLAVE_BYTES	False True Default: True	This entry can be used to specify that the bytes of the modules in the slave are to be swapped automatically by the driver. The driver only carries out this swapping in the case of digital modules; analog modules are not swapped. False: The bytes are not swapped. True: The bytes are swapped.
SLAVE_USED	False True Default: False	False: Slave is not initialized and no error messages are generated. True: Slave is initialized and error messages are generated.
CONTINUE_BY_ERR	False True Default: True	False: In the case of a fault in the slave ring, the application is notified that an error has occurred while reading and writing the I/Os (like in the case of a bus error). As soon as the fault in the slave ring has been rectified, this information is retracted. True: In the case of a fault, the application is not notified that an error has occurred while reading and writing the I/Os.

Entry	Value	Function
MASTER2SLAVE_OK_BIT	0 ... n Default: -1	This signals, in the higher-level ring, that the Interbus driver on the robot controller is still in the RUNNING state. This bit is not visible from the subordinate ring. This bit is reset as soon as the driver is in the RUNNING state. It is withdrawn if it is in a fault state or has not been started, and is not set, following a restart or a warm start, until the output data have been written. This bit cannot be be set or deleted manually from the robot controller. 0: Bit is not set. 1 ... n: Bit position in the I/O output memory after the start address of the slave.
SLAVE_CR	Default: 0	Communication reference of the slave if it is PCP-capable.

6.5 Configuring the bus with WorkVisual

Step	Description
1	Insert segments in the DTM catalog. (>>> 6.5.1 "Inserting segments in the DTM Catalog (Catalog Scan)" Page 28)
2	■ Configure the INTERBUS master. (>>> 6.5.2 "Configuring the INTERBUS master" Page 29) or ■ Configure the INTERBUS slave. (>>> 6.5.3 "Configuring the INTERBUS slave" Page 30) or ■ Configure the INTERBUS master and slave. (>>> 6.5.4 "Configuring the INTERBUS master and slave" Page 32)
3	Enter the IP address of the line interface (optional) (>>> 6.5.6 "Entering the IP address of the line interface" Page 34)

6.5.1 Inserting segments in the DTM Catalog (Catalog Scan)

Procedure

1. Open WorkVisual. **DTM Catalog Management** is opened.
2. Click on **Search for installed DTMs**. WorkVisual searches the PC for DTMs. The DTMs found are displayed.
3. Under **Known DTMs**, select the required DTMs and click on the **Right arrow** button.

   If all DTMs are to be accepted, click on the **Double right arrow** button.
4. The selected DTMs are displayed under **Current DTM Catalog**. Click on **OK**.

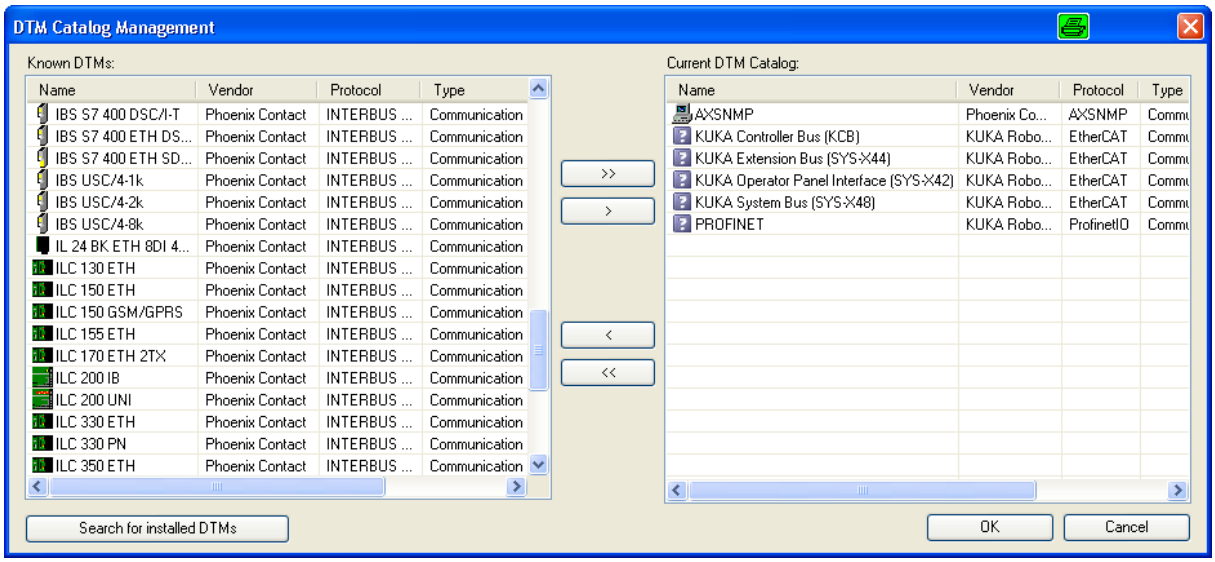


Fig. 6-3: DTM Catalog Management

6.5.2 Configuring the INTERBUS master

Description

The master bus configuration can be created directly with WorkVisual if no SVC file is being used. This means that, on booting, the master card detects the devices in sequence and determines the bus configuration automatically in this way. This procedure can only be applied if the devices are always available (no coupling and decoupling). Otherwise, the bus configuration must be created using Config+ (SVC file).

In both cases, the image (structure and sequence) of the Interbus must be replicated in WorkVisual. In the case of master configuration without the SVC file, it is sufficient to create the image with the device description files of the connected devices. In the case of master configuration with the SVC file, an offset must be entered, so that the precise addresses of the inputs and outputs are located at the right place in the image. Alternatively, placeholders can be inserted.

Precondition

- A robot controller has been added and set as active.

Procedure

1. Expand the tree structure of the robot controller on the **Hardware** tab in the **Project structure** window.
2. Right-click on **Bus structure** and select **Add…** from the context menu.
3. A window opens. Depending on which card is used, select the entry **IBS PCI SC/RI-I-T** or **IBS PCI SC/RI-LK** and confirm with **OK**. The entry is inserted in the tree structure.
4. Open the tree structure as far as possible. Right-click on **INTERBUS** and select **Add…** from the context menu. The **DTM Selection** window is opened.
5. Select the device used and confirm with **OK**. The device is inserted in the tree structure.
6. If necessary, repeat steps 4 and 5 for further devices.

> The device description files from the manufacturers or the generic device description files “KUKA Proxy” from KUKA can be used for the master configuration.
> The generic device description files “KUKA Slave Proxy” must not be used for the master configuration.

> The smallest possible memory unit is 2 bytes. A memory of 2 bytes is created in the image for 8 inputs and/or outputs.

Example

INTERBUS master configuration with SVC file:

A bus device has 32 inputs and outputs. The inputs and outputs are found at a specific address in the SVC file:

- 32 inputs at byte 10 (master ring)
- 32 outputs at byte 8 (master ring)

The master card creates the following image:

IN		OUT	
Empty 10 byte	32 inputs	Empty 8 byte	32 outputs

The image must be replicated in WorkVisual (>>> Fig. 6-4).

To ensure that the structure of the Interbus is read correctly, an offset must be entered so that the precise addresses of the inputs and outputs are located at the right place in the image (>>> 6.5.5 "Setting an offset" Page 33).

Alternatively, placeholders can be inserted, as shown in the example. All device description files except “KUKA Slave Proxy” can be used as placeholders in the master ring.

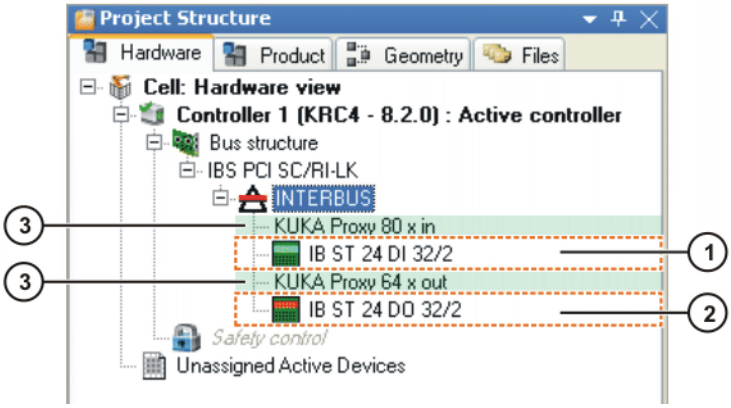


Fig. 6-4: Example of an INTERBUS master image

1	32 inputs	3	Placeholders (marked green)
2	32 outputs		

6.5.3 Configuring the INTERBUS slave

Description

The start address and the length of the image of the slave must be set in the Interbus communication DTM for configuration of the Interbus slave (>>> 6.4 "Configuring the file IBSPCI.XML" Page 24). Alternatively, the slave can also be configured with Config+ (SVC file) and the generated SVC file imported in the Interbus communication DTM. The image (bus configuration) must then be replicated in WorkVisual. The controller is inserted into the bus just like any other Interbus device.

Precondition

- A robot controller has been added and set as active.

Procedure

1. Expand the tree structure of the robot controller on the **Hardware** tab in the **Project structure** window.
2. Right-click on **Bus structure** and select **Add…** from the context menu.

3. A window opens. Depending on which card is used, select the entry **IBS PCI SC/RI-I-T** or **IBS PCI SC/RI-LK** and confirm with **OK**. The entry is inserted in the tree structure.
4. Open the tree structure as far as possible. Right-click on **INTERBUS** and select **Add…** from the context menu. The **DTM Selection** window is opened.
5. Select the device used and confirm with **OK**. The device is inserted in the tree structure.
6. If necessary, repeat steps 4 and 5 for further devices.

> The generic DTMs “KUKA Slave Proxy” from KUKA must be used for the slave configuration.

> The smallest possible memory unit is 2 bytes. A memory of 2 bytes is created in the image for 8 inputs and/or outputs.

Example

INTERBUS slave configuration with SVC file:

One bus device has 16 inputs and outputs, another has 48 inputs and outputs. These inputs and outputs are found at a specific address:

- 16 inputs at byte 896 (slave ring)
- 16 outputs at byte 896 (slave ring)

The slave card creates the following image:

IN		OUT	
Empty 896 byte	16 inputs	Empty 896 byte	16 outputs

> Creation of the image for the slave ring always starts at 896 bytes.

The image must be replicated in WorkVisual (>>> Fig. 6-5).

To ensure that the structure of the Interbus is read correctly, an offset of 896 must be entered so that the precise addresses of the inputs and outputs are located at the right place in the image (>>> 6.5.5 "Setting an offset" Page 33).

Alternatively, placeholders can be inserted. Since the creation of the image for the slave ring only starts at 896 bytes, a corresponding number of placeholders must be inserted. In this case, the following message placeholders have been inserted:

- 160 inputs and outputs (44x)
- 128 inputs and outputs (1x)

Only the device description files “KUKA Proxy” can be used as placeholders in the slave ring.

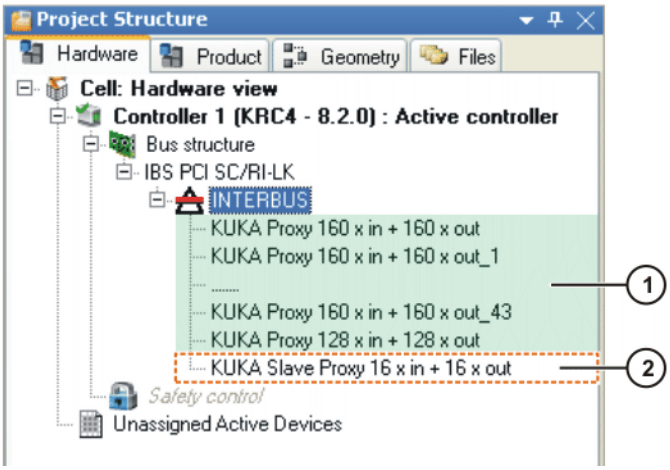


Fig. 6-5: Example of an INTERBUS slave image

1 Placeholders (marked green) 2 16 inputs and outputs

6.5.4 Configuring the INTERBUS master and slave

Description

In order to be able to configure the master and slave rings together, an SVC file must be created with Config+. The image (bus configuration) must then be replicated in WorkVisual.

Precondition

- A robot controller has been added and set as active.

Procedure

1. Expand the tree structure of the robot controller on the **Hardware** tab in the **Project structure** window.
2. Right-click on **Bus structure** and select **Add…** from the context menu.
3. A window opens. Depending on which card is used, select the entry **IBS PCI SC/RI-I-T** or **IBS PCI SC/RI-LK** and confirm with **OK**. The entry is inserted in the tree structure.
4. Open the tree structure as far as possible. Right-click on **INTERBUS** and select **Add…** from the context menu. The **DTM Selection** window is opened.
5. Select the desired device and confirm with **OK**. The device is inserted in the tree structure.
6. If necessary, repeat steps 4 and 5 for further devices.

> The device description files from the manufacturers or the generic device description files “KUKA Proxy” from KUKA can be used for the master configuration.
> The generic device description files “KUKA Slave Proxy” from KUKA must be used for the slave configuration.

> The smallest possible memory unit is 2 bytes. A memory of 2 bytes is created in the image for 8 inputs and/or outputs.

Example

INTERBUS master and slave configuration with SVC file:

A bus device has a master ring with 32 inputs and outputs and a slave ring with 16 inputs and outputs. These are found at a specific address:

- 32 inputs at byte 12 (master ring)
- 32 outputs at byte 8 (master ring)
- 16 inputs from byte 896 (slave ring)

- 16 outputs from byte 896 (slave ring)

The master and slave cards create the following image:

IN			
Empty 12 byte	32 inputs Master ring	Empty 880 byte	16 inputs Slave ring

OUT			
Empty 8 byte	32 outputs Master ring	Empty 884 byte	16 outputs Slave ring

> Creation of the image for the slave ring always starts at 896 bytes.

The image must be replicated in WorkVisual (>>> Fig. 6-6).

To ensure that the structure of the Interbus is read correctly, an offset of 896 must be entered so that the precise addresses of the inputs and outputs are located at the right place in the image (>>> 6.5.5 "Setting an offset" Page 33).

Alternatively, placeholders can be inserted. Since the creation of the image for the slave ring only starts at 896 bytes, a corresponding number of placeholders must be inserted. In this case, the placeholder with the 160 inputs/outputs has been inserted 44 times. Only the device description files “KUKA Proxy” can be used as placeholders in the master and slave ring.

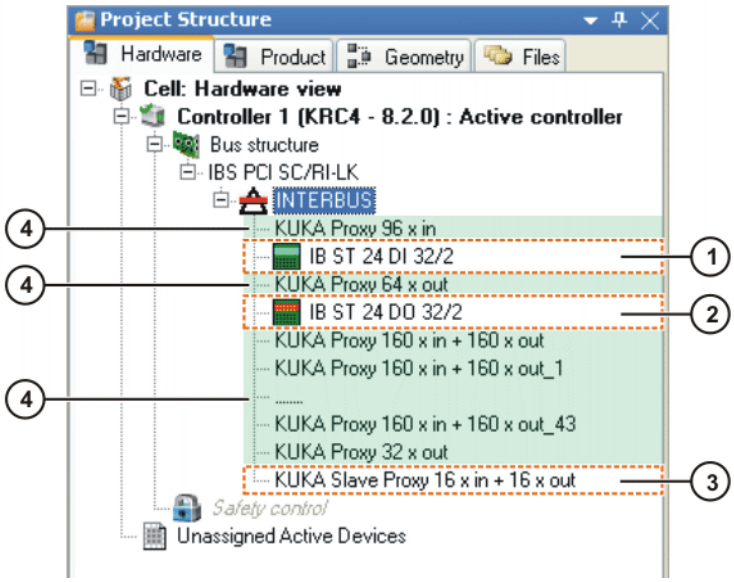


Fig. 6-6: Example of an INTERBUS master and slave image

1	32 inputs on master	3	16 inputs and outputs on slave
2	32 outputs on master	4	Placeholders (marked green)

6.5.5 Setting an offset

Precondition

- The image is replicated in WorkVisual.

Procedure

1. Expand the tree structure of the robot controller as far as possible on the **Hardware** tab in the **Project structure** window.
2. Right-click on the device used and select **Settings…** from the context menu. The **Device settings** tab is displayed.

3. Enter the desired offset and confirm it by pressing **OK**. The device is assigned to the specified address.

> Entering “0” as the address allows the devices to be concatenated in the image. This means that the devices can also be offset as a block by changing only the address of the 1st device.

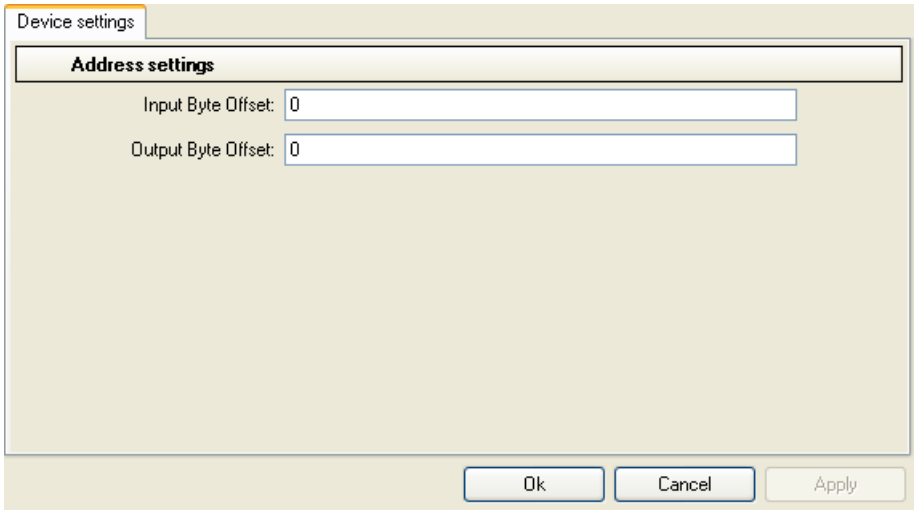


Fig. 6-7: Device settings tab

6.5.6 Entering the IP address of the line interface

Description

In order to be able to perform online functions, e.g. a bus scan, the IP address of the line interface must be entered.

Precondition

- A robot controller has been added and set as active.
- **IBS PCI SC/RI-I-T** or **IBS PCI SC/RI-LK** is added.

Procedure

1. Expand the tree structure of the robot controller on the **Hardware** tab in the **Project Structure** window.
2. Depending on which card is used, right-click on **IBS PCI SC/RI-I-T** or **IBS PCI SC/RI-LK** and select **Settings...** in the pop-up menu. The **Master Settings** window opens.
3. Enter the IP address of the line interface and confirm with **OK**.

7 Operation

7.1 Coupling/decoupling segments

Description

For certain applications, e.g. tool change, it is necessary to couple and decouple segments. Coupling and decoupling can be carried out via the HMI.

Decoupling

Properties of decoupled segments:

- If decoupled segments are disconnected from INTERBUS or the power supply, no error is triggered.
- All I/O operations on decoupled segments remain without effect.
- Decoupled segments cannot carry out error treatment in the case of read/write errors.

Coupling

The IOCTL function has a blocking effect. It only returns when the coupling operation has been executed and the response from the firmware can be returned. In the case of a positive response, the segment can be used at once. If a negative response is returned, an error has occurred during coupling.

If a coupled device is not functional, e.g. because it is disconnected from the bus or supply voltage, a message is displayed.

7.1.1 Coupling/decoupling segments via HMI

Procedure

1. Select the menu sequence **Display** > **Variable** > **Single**.
2. In the **Name** box, enter:
   - To decouple: =IOCTL("IBS1",60,*[Segment number]*)
   - To couple: =IOCTL("IBS1",50,*[Segment number]*)
3. Confirm by pressing the Enter key. The segment is coupled or decoupled.

> Only segments that are present in the network configuration can be coupled or decoupled.

If a segment cannot be coupled or decoupled, the IOCTL command returns a negative value.

If a segment is successfully coupled or decoupled, the IOCTL command returns the number of the segment.

> Further information about this IOCTL command can be found here: (>>> 8.5.3 "Switching segments on and off" Page 43).

7.1.2 Coupling/decoupling segments via KRL

Syntax

Decoupling:

```
RET =IOCTL("IBS1",60,Segment number)
```

Coupling:

```
RET =IOCTL("IBS1",50,Segment number)
```

Example

Here, segment 512 is decoupled, depending on the tool used.

```
...
IF (NEXT_TOOL == GRIPPER_1) THEN
  RET = IOCTL("IBS1",60,512)
ENDIF
...
```

7.2 Activating/deactivating the Interbus driver

Description

The Interbus driver can be activated and deactivated via the HMI.

The Interbus driver is active once it has been installed.

Precondition

- "Expert" user group

Procedure

1. In the main menu, select **Configuration** > **Inputs/outputs** > **I/O driver**.
2. Activate/deactivate the Interbus driver:
   - Activate:
     Set the check mark in the "Installed" column after the Interbus name.
   - Deactivate:
     Remove the check mark in the "Installed" column after the Interbus name.
3. Press the **Close** icon. Answer the request for confirmation with **Yes**. Reconfiguration is carried out.

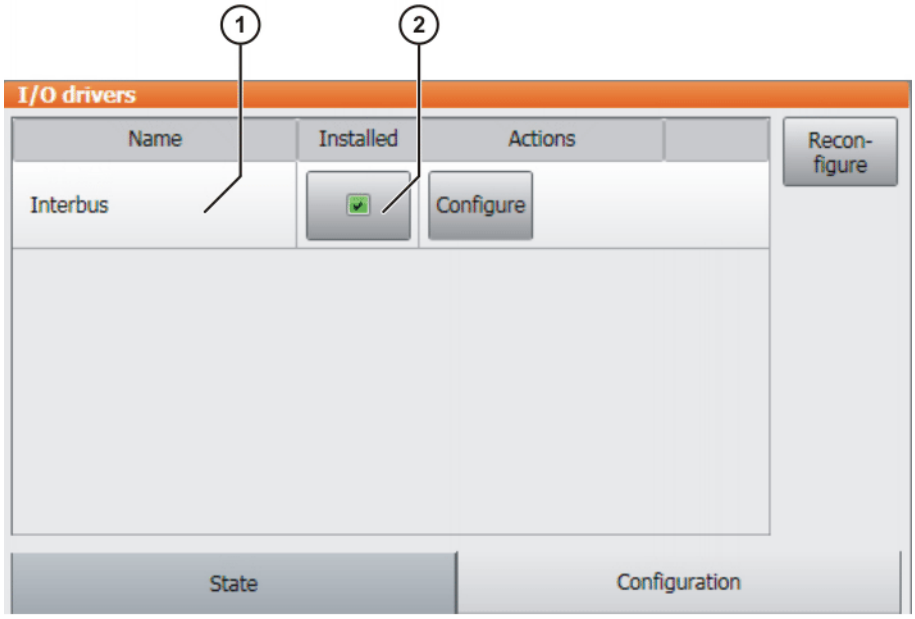


Fig. 7-1: Activating/deactivating the Interbus driver

1 Interbus name
2 "Installed" column

8 Diagnosis

8.1 Displaying diagnostic data

Procedure

1. Select the menu sequence **Diagnosis** > **Diagnostic Monitor**.
2. In the **Module** box, select the entry **Interbus driver (IBusDrv)**.
   Diagnostic data are displayed.

Description

“Name” column	Description
version of driver	Version of the driver
name of bus instance	Name of the bus instance
state of interbus master	Status of the Interbus master
State Interbus Slave	Status of the Interbus slave Only displayed if the slave is active.
Parameter from ibspci.xml used	■ **0**: Parameters from IBSPCI.XML have not been read. ■ **1**: Parameters from IBSPCI.XML have been read.
Bus error by device number	Indicates which device has a bus error.
Error code from the bus error	Displays the error code and the corresponding bus error.
Bus failure by device number	Indicates which device has a bus fault.
Last segment activated	The number indicates the most recently activated segment.
last segment deactivated	The number indicates the most recently deactivated segment.
Error code from switching on/off last segment	Displays the error code of the last switched segment.
State register	Value of the status register
Parameter register	Value of the parameter register
Extended parameter register	Value of the extended parameter register
Current PD cycle time	Current PD cycle time
Permitted PD cycle time	Specified PD cycle time
Transfer frequency quality bit is set	■ **0**: Transfer quality bit has been activated. ■ **1**: Transfer quality bit has not been activated.
Diagnostic register shows faulty data cycle bit	■ **0**: Diagnostic register displays no faulty data cycle bit. ■ **1**: Diagnostic register displays faulty data cycle bit.
state of state machine from driver	Status information of the driver (>>> 8.2 "Status information of the driver" Page 38)

“Name” column	Description
IP address of configuration tool (1)	IP address of a connected configuration tool (1)
IP address of configuration tool (2)	IP address of a connected configuration tool (2)

8.2 Status information of the driver

The “State of status machine of driver” can be read in the diagnostic monitor.

Designation	Value	Description
INST_NOT_AVAILABLE	0x0001	Initial state if the instance has not yet been initialized.
INST_RUNNING	0x0002	This status bit indicates that the Interbus was started successfully and that I/O data exchange can take place.
INST_STOP_USER	0x0004	This status bit is set if the Interbus has been stopped by the driver or by a user action.
INST_STOP_ERR	0x0008	This status bit is set if an error has occurred in the Interbus master ring, preventing an update of all I/Os, e.g. in the case of a bus error.
INST_CLEAR_MPM_OUT	0x0010	This status bit is set if the outputs are set to 0 in the event of an error.
INST_WAITING_FOR_EX_START	0x0020	This status bit is set if the Interbus is to be started by means of a boot project or “externally” and is waiting for this start.
INST_STOP_WARN	0x0040	This status bit is set if the driver is to signal an error in the case of a periphery fault when writing and reading I/Os.
INST_RUN_WATCHDOG_START	0x0080	This status bit is set if the Interbus has already been started, but the watchdog is still to be triggered. The watchdog is not triggered until the cyclical read and write function is executed. The start of the watchdog must be delayed until the cyclical read and write function has been activated.
INST_STOP_FREEZE	0x0100	This state is set once the driver has been set to the Freeze state.
INST_STOP_RESET	0x0200	This status bit is set if the driver attempts to restart the Interbus.
INST_STOP_WATCHDOG	0x0400	This status bit is set if the watchdog has expired.
INST_STOP_HW_ERR	0x0800	This status bit is set if a hardware error has been detected or signaled by the firmware.
INST_STOP_SLV_ERR	0x1000	This status bit is set if the driver is no longer to update I/O data once the slave ring has signaled an error.
INST_ERR_DETECTING	0x2000	This status bit indicates that the firmware has started searching for an error in the Interbus master ring.

Designation	Value	Description
INST_AUTO_RESTART	0x4000	This status bit indicates that the driver should cyclically attempt to restart the bus in the case of a bus error. The bus is thus automatically restarted once the error has been rectified.
INST_DOING_SHUTDOWN	0x8000	This status bit indicates that the driver is being shut down. This bit is only set in addition to all the others and serves to provide information for other functions (e.g. IOCTL) that must be terminated.

8.3 LEDs on the master module

The diagnostic LEDs on the master module are all accommodated on the front plate. The basic functions of the module are checked when it is switched on. If no faults are detected, “SC” flashes green after approx. 5 seconds. “HF” goes out when the drivers are activated.

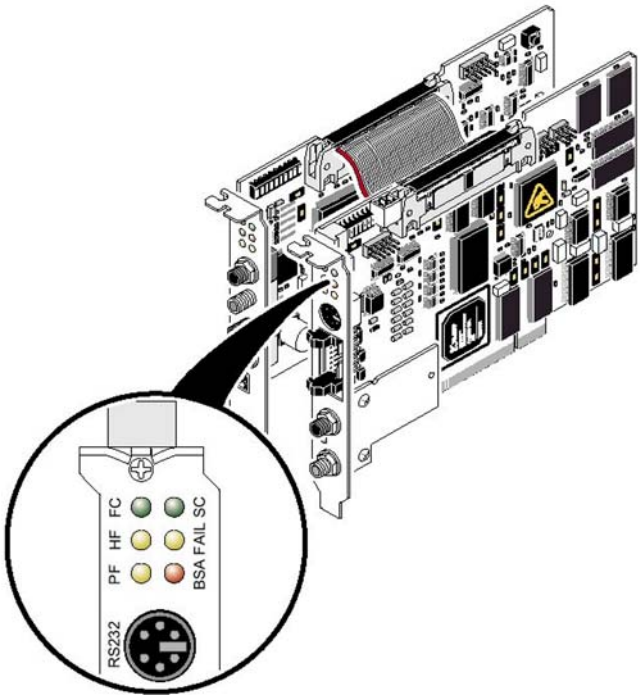


Fig. 8-1: LEDs on the master module

The state of the Interbus is indicated by other LEDs:

Designation	Color	Meaning
FC	Green	Reserved
SC RDY/RUN	Green	Interbus Ready The controller board has the state READY or ACTIVE
SC RDY/RUN	Flashing	Interbus Running The controller board has the state RUN
HF	Yellow	Host Failure System fault of the host
FAIL	Red	Failure A fault has occurred in the Interbus system

Designation	Color	Meaning
PF	Yellow	Peripheral Failure Periphery fault of a device
BSA	Yellow	Bus Segment Aborted One or more bus segments are switched off

The CMD interface is designed as a 6-contact mini-DIN socket (PS/2) on the front plate .

The master module also has an LED FO3 (Fiber Optic 3) for diagnosis of the outgoing fiber-optic cable.

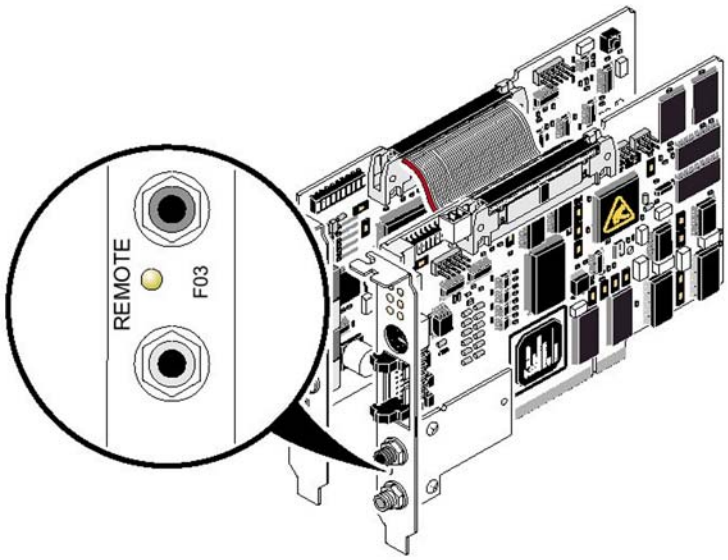


Fig. 8-2: LED for diagnosis of the outgoing fiber-optic cable interface

Designation	Color	Meaning
FO3	Yellow	Fiber Optic 3 Lights up when the initialization of the outgoing interface is not OK, or a MAU warning is present due to poor transmission quality on the path. This applies to the outgoing data path/transmitter to the following module; the state of the return data path/receiver is diagnosed by the following module.

8.4 LEDs on the slave module

The diagnostic LEDs of the slave module indicate its state and that of the higher-level Interbus system:

Designation	Color	Meaning
UL	Green	U Logic Operating voltage present
RC	Green	Remote bus Check The connection to the higher-level controller board has been established
BA	Flashing	Bus Active Bus is in the ACTIVE state

Designation	Color	Meaning
BA	Green	Bus Running Bus is in the RUN state
RD	Red	Remote bus Disabled The outgoing remote bus interface is deactivated

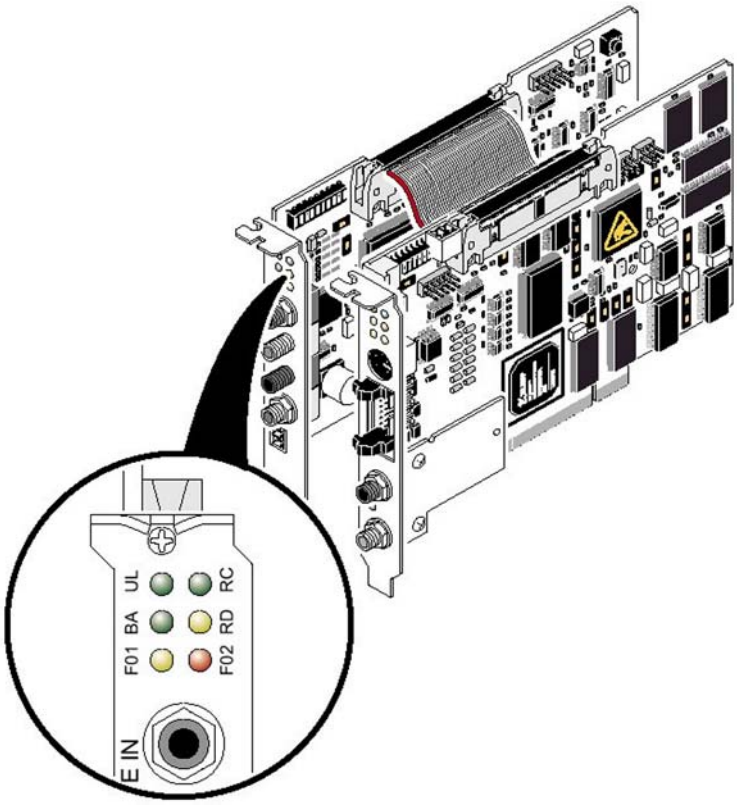


Fig. 8-3: LEDs on the slave module

The slave module also has two other LEDs for diagnosis of the fiber-optic cable:

Designation	Color	Meaning
FO1, FO2	Yellow	Fiber Optic 1, Fiber Optic 2 Light up when the initialization of the outgoing interface is not OK, or a MAU warning is present due to poor transmission quality on the path. This applies to the outgoing data path/transmitter to the following module; the state of the return data path/receiver is diagnosed by the following module.

8.5 IOCTL commands

Description The IOCTL commands are also used for advanced troubleshooting.

Precondition
- All communications cables have been installed.
- Expert user group

Procedure

IOCTL commands can be entered via TELNET or KRL.

- TelNet shell:
  RETURN VALUE = iosysIoctl (Instance name, REQUEST, Parameter)
- KRL interface:
  RETURN VALUE = IOCTL (Instance name, REQUEST, Parameter)

Parameter	
Return value	Depends on the function (REQUEST number) called.
Instance name	The name of the instance can be found in the Interbus.XML file (parameter busInstanceName).

Overview of functions

Entry	Request	Parameter	Function
CP_IB_DUMP	1012	-	Creation of a dump file (MPM log file)
IODRV_IOCTL_RESTART	12	-	Interbus is stopped, reconfigured and restarted.
IODRV_IOCTL_ACTIVATE_DEVICE	50	Segment	Activation of alternative groups
IODRV_IOCTL_DEACTIVATE_DEVICE	60	Segment	Deactivation of alternative groups
IODRV_IOCTL_PRINT_INFO	100	-	The diagnosis and the parameter register of the master and slave are output to the telnet.
CP_IB_SEND_QUIT_PF	1017	-	Sends the message “Acknowledge periphery faults of all devices” to the firmware. The system does not wait for a response from the firmware indicating whether the command could be executed successfully.
CP_IB_GET_SLAVE_STATE	1011	-	Contents of the status register of the slave module
CP_IB_GET_DEVICE_STATE	1013	Segment, position	Current state of a module

8.5.1 Generating a dump file

`CP_IB_DUMP`

The name of the dump file can be specified in the entry DUMPFILE in the [CONFIGURATION] section of IBSPCI.XML. In this case, all write access operations are recorded on the MPM. The ibsPciDump function can be executed by means of the IOCTL command `CP_IB_DUMP`.

8.5.2 Restart

`IODRV_IOCTL_RESTART`

The Restart command attempts to restart the controller board by means of the firmware command `Start_Data_Transfer_Request`. No restart is carried out if the controller board is still active (even in the event of a periphery fault). An error message to this effect is sent to the KUKA.HMI and the function is exited with an error code (ERROR). This informs the user that the controller board is active, but that there is still a periphery fault.

If an error occurs during the restart, the corresponding error treatment is carried out.

8.5.3 Switching segments on and off

`IODRV_IOCTL_ACTIVATE_DEVICE,`
`IODRV_IOCTL_DEACTIVATE_DEVICE`

These IOCTL commands can be used to activate and deactivate alternative groups (segments). In the event of a warm start, the Interbus driver saves the last active segment during the shut-down procedure and automatically activates it again on rebooting. In the case of a cold start, the segment to be activated (if any) must be communicated to the Interbus by means of an IOCTL command from the kernel system. In the event of an error, the IOCTL command returns the following values:

Message	Value	Description
ERROR	-1	Unable to send message
IBS_PCI_SWITCH_WRONG_MODULE	-2	Invalid segment number
IBS_PCI_TRY_SWITCH_ON_SECOND_SEGM	-3	Attempt to switch on a second alternative segment
IBS_PCI_TRY_SWITCH_OFF_FIRST_SEGM	-4	Attempt to switch off the first device
IBS_PCI_NO_EXCLUSIVE_RIGHTS	-5	No exclusive rights for the service
IBS_PCI_SWITCH_GROUP_ERR	-6	Group conflict when devices switched on or off
IBS_PCI_SWITCH_MODULE_ERR	-7	Device conflict when devices switched on or off
IBS_PCI_BLOCKING_COMMANDO	-8	A service that is already active is blocking execution of this service.
IBS_PCI_UNKNOWN_ERR	-100	Unknown error

If the segment was switched correctly, the command returns the number of the switched segment.

8.5.4 Extended state polling of slave

`CP_IB_GET_SLAVE_STATE`

As the slave can be operated independently of the master, the slave also has its own states. The read and write functions are the same for the master and slave.

The slave can have the following states:

- Slave Data Transfer (bit 1)
- Fail (bit 2)
- Slave Initialized (bit 3)
- Power On (bit 4)
- Ready (bit 5)

8.5.5 Polling the state of a module

`CP_IB_GET_DEVICE_STATE`

This IOCTL command can be used to poll the state of a module. The return value given by this command is the state of the module. This command must be sent to the driver along with the number of the module in the form of the segment and position number.

A device (module) can have the following states:

- Alarm Output (bit 0)
- Error Output (bit 1)
- MAU detection of the incoming remote bus interface (towards data ring) (bit 9)
- MAU detection of the incoming remote bus interface (away from data ring) (bit 10)
- Periphery Fail (bit 11)

All other bits in the word are reserved.

A detailed description of the state information can be found in the Interbus documentation from Phoenix Contact under the firmware command `Read_Device_State_Request`.

8.6 PCP functionality of the slave

Unambiguous remote addresses are determined for each PCP device in the higher-level ring for the purpose of managing the services PCP (parameter data channel) and PNM7 (remote management utility). Unlike the local communication reference (CR) of the lower-level ring (master ring), the remote address issued is unambiguous in the network. Each PCP-capable device is referenced in the higher-level ring by means of the position of the data in the summation frame protocol. The CRs in the master ring are issued in ascending order, as a continuous series starting with 2. 2 CRs can be issued for an Interbus controller board: one each for PCP and PNM7.

Name	Abbreviation	Function	
REQUEST	REQ	Request for a service	0xxxhex
INDICATION	IND	Receipt of the service request	4xxxhex
RESPONSE	RES	Response to the service request	Cxxxhex
CONFIRMA-TION	CON	Confirmation of the service	8xxxhex

PCP messages that are evaluated by the driver have the identifier 40xxhex. Before further processing, the received CR is compared with the preset CR in the driver to check that they match.

8.6.1 PCP hardware settings

See (>>> 6.3 "DIP switches on the slave module" Page 22).

8.6.2 Establishing the connection

`PCP_INITIATE_IND_CODE`

If the driver receives a message with the command `PCP_INITIATE_IND_CODE`, a positive response message is returned. The parameters **Access_Group** and **Password** are ignored and set to 0 in the response message.

8.6.3 Reading data

`PCP_READ_IND_CODE`

If a message is received with the command `PCP_READ_IND_CODE`, a check is made to see if the PCP object 0x5FFF is involved. If the message contains a different PCP object, a negative response message is returned with the error code 0x0607 (Index not supported). If the message contains a PCP object with the index 0x5FFF, the subindex is evaluated according to the two following tables.

- If the subindex is between 1 and 32, then 128-byte data are read from the input range of the MPM and these data are returned in a positive response.
- If the subindex is between 34 and 66, then 128-byte data are read from the output range of the MPM and these data are returned in a positive response.
- If the subindex is 0, data are read from the input range of the MPM, using the write service with subindex 1 according to the settings that have been made, and these data are returned with a positive response.
- Subindex 33 functions in the same way as subindex 0, except that in this case the data are read from the output range of the MPM.

Index	Subindex	MPM range	Length	Access
0x5FFF	0	Can be set using object 5FFE subindex 1	Variable, max. 240 bytes	Read-only
0x5FFF	1	0 ... 127	128 bytes	Read-only
0x5FFF	2	128 ... 255	128 bytes	Read-only
0x5FFF	3	256 ... 383	128 bytes	Read-only
0x5FFF	4	384 ... 511	128 bytes	Read-only
0x5FFF	5	512 ... 639	128 bytes	Read-only
0x5FFF	6	640 ... 767	128 bytes	Read-only
0x5FFF	7	768 ... 895	128 bytes	Read-only
0x5FFF	8	896 ... 1023	128 bytes	Read-only
0x5FFF	9	1024 ... 1151	128 bytes	Read-only
0x5FFF	10	1152 ... 1279	128 bytes	Read-only
0x5FFF	11	1280 ... 1407	128 bytes	Read-only
0x5FFF	12	1408 ... 1535	128 bytes	Read-only
0x5FFF	13	1536 ... 1663	128 bytes	Read-only
0x5FFF	14	1664 ... 1791	128 bytes	Read-only
0x5FFF	15	1792 ... 1919	128 bytes	Read-only
0x5FFF	16	1920 ... 2047	128 bytes	Read-only
0x5FFF	17	2048 ... 2175	128 bytes	Read-only
0x5FFF	18	2176 ... 2303	128 bytes	Read-only
0x5FFF	19	2304 ... 2431	128 bytes	Read-only

Index	Subindex	MPM range	Length	Access
0x5FFF	20	2432 ... 2559	128 bytes	Read-only
0x5FFF	21	2560 ... 2687	128 bytes	Read-only
0x5FFF	22	2688 ... 2815	128 bytes	Read-only
0x5FFF	23	2816 ... 2943	128 bytes	Read-only
0x5FFF	24	2944 ... 3071	128 bytes	Read-only
0x5FFF	25	3072 ... 3199	128 bytes	Read-only
0x5FFF	26	3200 ... 3327	128 bytes	Read-only
0x5FFF	27	3328 ... 3455	128 bytes	Read-only
0x5FFF	28	3456 ... 3583	128 bytes	Read-only
0x5FFF	29	3584 ... 3711	128 bytes	Read-only
0x5FFF	30	3712 ... 3839	128 bytes	Read-only
0x5FFF	31	3840 ... 3967	128 bytes	Read-only
0x5FFF	32	3968 ... 4095	128 bytes	Read-only

MPM input data range from address 1000hex

Index	Subindex	MPM range	Length	Access
0x5FFF	33	Can be set using object 5FFE subindex 1	Variable, max. 240 bytes	Read-only
0x5FFF	34	0 ... 127	128 bytes	Read-only
0x5FFF	35	128 ... 255	128 bytes	Read-only
0x5FFF	36	256 ... 383	128 bytes	Read-only
0x5FFF	37	384 ... 511	128 bytes	Read-only
0x5FFF	38	512 ... 639	128 bytes	Read-only
0x5FFF	39	640 ... 767	128 bytes	Read-only
0x5FFF	40	768 ... 895	128 bytes	Read-only
0x5FFF	41	896 ... 1023	128 bytes	Read-only
0x5FFF	42	1024 ... 1151	128 bytes	Read-only
0x5FFF	43	1152 ... 1279	128 bytes	Read-only
0x5FFF	44	1280 ... 1407	128 bytes	Read-only
0x5FFF	45	1408 ... 1535	128 bytes	Read-only
0x5FFF	46	1536 ... 1663	128 bytes	Read-only

Index	Subindex	MPM range	Length	Access
0x5FFF	47	1664 ... 1791	128 bytes	Read-only
0x5FFF	48	1792 ... 1919	128 bytes	Read-only
0x5FFF	49	1920 ... 2047	128 bytes	Read-only
0x5FFF	50	2048 ... 2175	128 bytes	Read-only
0x5FFF	51	2176 ... 2303	128 bytes	Read-only
0x5FFF	52	2304 ... 2431	128 bytes	Read-only
0x5FFF	53	2432 ... 2559	128 bytes	Read-only
0x5FFF	54	2560 ... 2687	128 bytes	Read-only
0x5FFF	55	2688 ... 2815	128 bytes	Read-only
0x5FFF	56	2816 ... 2943	128 bytes	Read-only
0x5FFF	57	2944 ... 3071	128 bytes	Read-only
0x5FFF	58	3072 ... 3199	128 bytes	Read-only
0x5FFF	59	3200 ... 3327	128 bytes	Read-only
0x5FFF	60	3328 ... 3455	128 bytes	Read-only
0x5FFF	61	3456 ... 3583	128 bytes	Read-only
0x5FFF	62	3584 ... 3711	128 bytes	Read-only
0x5FFF	63	3712 ... 3839	128 bytes	Read-only
0x5FFF	64	3840 ... 3967	128 bytes	Read-only
0x5FFF	65	3968 ... 4095	128 bytes	Read-only

MPM output data range from address 0000hex

8.6.4 Writing data

`PCP_WRITE_IND_CODE`

If a message is received with the command `PCP_WRITE_IND_CODE`, a check is made to see if the PCP object 0x5FFE is involved. If the message contains a different PCP object, a negative response message is returned with the error code 0x0607 (Index not supported). If the message contains a PCP object with the index 0x5FFE, the subindex is evaluated according to the above tables. If the subindex is 1 or 2, a check is made to see if the specified range is valid.

In the MPM, 4096 bytes are saved for the input data and 4096 bytes for the output data. The start address can thus be selected between 0 and 4096. Here also, a check is made to see if the start address plus the length of the data to

be read is less than 4096. If the check of the start address and length fails, a negative response is returned with the error code 0x0605 (Application error). If the specified data are correct, they are accepted and a positive response is returned.

Index	Subindex	Meaning of user data	Length	Default	Access
0x5FFF	1	Byte 0: MPM In start address (high) Byte 1: MPM In start address (low) Byte 2: Length of the MPM range	3 bytes	0 0 240	Read/ Write
0x5FFF	2	Byte 0: MPM Out start address (high) Byte 1: MPM Out start address (low) Byte 2: Length of the MPM range	3 bytes	0 0 240	Read/ Write

Configuration objects for variable access

If the subindex of the PCP message is 10, user data are to be written to the MPM output range. Before the data are written to the MPM, the specified address range is checked. If the limit of 4096 bytes is to be exceeded, the data are not accepted and a negative response is returned with the error code 0x0605 (Application error). If the address range is OK, the data are written to the MPM accordingly and a positive response is returned.

Index	Subindex	Meaning of user data	Length	Access
0x5FFF	10	Byte 0: MPM Out start address (high) Byte 1: MPM Out start address (low) Byte 2: User data 1 ... Byte n: User data n-1	Variable, max. 240 bytes	Read/Write

Writing objects for user data

8.6.5 Terminating the connection

The PCP messages PCP_ABORT_IND_CODE and PCP_REJECT_IND_CODE are received, but the driver does not react to these messages.

8.6.6 PCP connection settings

When the Interbus is started, the driver checks whether the controller board slave supports PCP. If so, the PDU size of PCP objects 0x5FFF and 0x5FFE is changed to 246 bytes.

8.6.7 PCP server response to a fault in the master ring

If the Interbus driver is switched to the inactive state as the result of a fault on the Interbus, all incoming PCP messages receive a negative response. The Error_Class and Error_Code word are set to 0x0902.

9 Messages

9.1 KUKA.HMI error messages

Bus error and periphery fault messages are implemented as “status messages” in the KUKA.HMI. In this way, messages are withdrawn by the driver when the faults have been eliminated or a different error message is present. Error codes and additional information generated with the messages come from the firmware.

Further information about the meaning of these codes can be found in chapter 5.2 of the Phoenix Contact documentation “Interbus User Manual – Firmware Services and Error Messages – IBS SYS FW G4 UM”.

In the case of an error, the Interbus driver signals the error to the application in accordance with the settings in the file IBSPCI.XML. It now depends on the application how it reacts to the error, e.g. the KUKA.HMI can generate a read/write error and interrupt the execution of KRL programs.

%1, %2 and %3 are individual variables of the error messages. %1 is always the instance name of the driver.

Message	Reason	Effect	Remedy
%1 Bus error %2 segment %3	Bus error with specification of the device in which the error was localized. The error code (%2) is saved by the firmware. Further information can be found in the Phoenix Contact documentation.	No input and outputs data are exchanged with the controller board.	Eliminate bus error and restart the driver.
%1 user error: %2 additional info: %3	An operator error has been made by the user, e.g. incorrect parameter when calling a firmware service. The error code (%2) is saved by the firmware. Further information can be found in the Phoenix Contact documentation.	The effect depends on the specific error. Further information can be found in the description of the error code.	Information about remedial action can be found in the description of the error code.
%1 System error: %2 additional info: %3	A system error has occurred, probably in the hardware. The error code (%2) is saved by the firmware. Further information can be found in the Phoenix Contact documentation.	No input and outputs data are exchanged with the controller board.	Information about remedial action can be found in the description of the error code. It may be necessary to exchange the controller board.

Message	Reason	Effect	Remedy
%1 Current configuration is not identical to active configuration	The BSA_BIT in the diagnostic register has been set, i.e. the current configuration does not match the active configuration.	The current configuration cannot be loaded.	Adapt current configuration or delete active configuration and then start the current configuration.
%1 Bus error. Error location in progress.	The firmware has discovered a bus error and is now searching for the cause.	No input and outputs data are exchanged with the controller board.	Wait until the firmware has found the cause of the error.
%1 diagnostic register shows faulty data cycle bit.	The “faulty data cycles” bit in the diagnostic status register has been set. This is only used in synchronous mode.	Information about the effect can be found in the description of the diagnostic status register of the controller board.	Information about the effect can be found in the description of the diagnostic status register of the controller board.
%1 Error opening file %2.	An error occurred when opening the specified file.	The driver cannot be started.	Check whether the file is present in the correct directory or whether it has an incorrect format.
%1 No restart carried out, as no error present.	The restart is not executed, as the Interbus is running or has not yet ben started.	Restart is not carried out.	If the driver is to be restarted, carry out an I/O reconfiguration.
%1 Periphery failure, device number %2.	A periphery fault has occurred in the device.	If the CONTINUE_WITH_WARNING flag has been set to 0, the exchange of input and output data is stopped.	Eliminate periphery fault.
%1 Bus error in slave circuit.	There is a bus error in the slave ring or the slave ring has not yet been started.	No data are exchanged with the slave ring. If CONTINUE_BY_ERR has been set to 0 in the INI file of the driver, no more input and output data are exchanged with the master ring.	Eliminate bus error in slave ring or start slave ring.
%1 error in slave circuit %2, additional info %3.	There is an error in the slave part of the controller board. The error code (%2) and additional information come from the firmware and describe the error.	There is an error in the slave part of the controller board. The error code and additional information come from the firmware and describe the error.	Eliminate bus error in slave ring or start slave ring.

Message	Reason	Effect	Remedy
%1 Error %2 switching off segment %3.	An error occurred when switching a segment off. The error code (%2) is saved by the firmware. Further information can be found in the Phoenix Contact documentation.	The effect depends on the specific error. Further information can be found in the description of the error code.	Information about remedial action can be found in the description of the error code.
%1 Error %2 switching on segment %3.	An error occurred when switching a segment on. The error code (%2) is saved by the firmware. Further information can be found in the Phoenix Contact documentation.	The effect depends on the specific error. See description of the error codes.	Information about remedial action can be found in the description of the error code.
%1 Synchronization error.	A synchronization error has occurred in “bus synchronous” mode.	No input and outputs data are exchanged with the controller board.	Check the system or set the PD cycle time to a higher value using the “Set_Value” service (0750hex).
%1 bus error %2, additional info %3.	A bus error has occurred. The error code (%2) and additional information are saved by the firmware. Further information can be found in the Phoenix Contact documentation.	No input and outputs data are exchanged with the controller board.	Eliminate bus error and reset the driver.
%1 transfer quality bit has been activated.	The “specified error density exceeded” bit in the diagnostic status register has been set.	Not all data cycles are correctly executed.	Check bus structure.
%1 Error in status register %2, parameter register %3.	Indicates an error about which the driver has no further information. A description of the status and parameter registers can be found in the Phoenix Contact documentation.	The effect depends on the specific error. See description of the registers. No input and outputs data are exchanged with the controller board.	See description of the registers.
%1 The watchdog has expired.	The SYS_FAIL bit in the diagnostic status register has been set.	The effect depends on the application (see note above table).	Carry out I/O reconfiguration, reboot controller, exchange controller board.

Message	Reason	Effect	Remedy
%1 Restart already in progress.	A restart is being carried out (possibly automatically).	No input and outputs data are exchanged with the controller board.	If the automatic restart has been activated in the XML file of the driver, deactivate it again.
%1 Error accessing controller board.	Possible causes: ■ Controller board is defective and must be exchanged. or ■ The board number in the XML file does not match the board number set on the controller board.	The driver cannot initialize the controller board. Communication with the controller board is not possible.	Depending on the cause, either the controller board must be exchanged or the board number in the XML file must be adapted.
%1 waiting for external start of Interbus.	The Interbus driver has been set to true with the “EXTERN_START” flag and is now waiting for the Interbus to be started externally, e.g. using the CMD tool.	No I/O data are executed with the Interbus.	Either set “EXTERN_START” flag to false, start the Interbus via a tool or use a boot project.
%1 Caution! Bus mode is not 'Asynchronous with synchronization pulse'.	Notification message that the Interbus is not in “Asynchronous with synchronization pulse” mode.	Correct functioning of the Interbus driver is only assured in “Asynchronous with synchronization pulse” mode. In the other modes, it is possible, for example, that data may not be written to the outputs.	Switch the bus mode back to “Asynchronous with synchronization pulse”.
%1 slave address unknown.	The driver was unable to determine the address of the controller board slave. Either it has not been configured, or the slave could not be started.	The driver is not started, as it is not possible to access the I/O data of the slave.	Configure the address of the slave, e.g. in the CMD tool.
%1 slave ID on card (%2) is different from that in XML file (%3).	The slave ID on the card is configured differently from that specified in the XML file of the driver.	If the higher-level master expects a different slave ID, a bus error will be generated in the higher-level ring.	Either the ID in the XML file is incorrect, or the slave has been incorrectly configured (e.g. via the DIP switches).
%1 Version mismatch in the file %2, required version is %3.	The file that has been read has a different identifier from that required by the current software.	The data from the file cannot be used.	Use the corresponding version of the file and adapt the data in the new file if necessary.

Message	Reason	Effect	Remedy
%1 File %2 cannot be read (formatting error).	The format of the file is incorrect, resulting in the parsing of the file being canceled.	The data from the file cannot be used. The default values are used.	Check the format of the file. See also the corresponding XSD file.
%1 no master or slave circuit activated.	Neither the master nor the slave ring of the Interbus card has been activated in the IBSPCI.XML file.	No connection is established to the Interbus card. Communication with the Interbus card is not possible.	Activate either the master or the slave or both.
%1 Loading of SVC file aborted due to a formatting error.	The format of the command in the SVC file is incorrect in at least one line.	The loading of the command from the SVC file is aborted. Depending on where this cancellation occurs, the Interbus remains in the corresponding state. The driver continues to boot, but without I/O data traffic.	Correct or re-generate the SVC file.
%1 Loading of SVC file aborted due to an error in a firmware service	An error occurred during transmission of a firmware service from the SVC file. Either the firmware service is not correctly specified, or it could not be executed, or it generates an error.	The loading of the command from the SVC file is aborted. Depending on where this cancellation occurs, the Interbus remains in the corresponding state. The driver continues to boot, but without I/O data traffic.	Check whether the firmware service is correctly specified. The bus configuration does not match the configuration that is to be loaded with the SVC file.
%1 Error in connection to firmware.	No connection can be established to the message interface of the firmware. The maximum number of connections has been reached or the firmware is in an error state and can no longer be addressed.	No message communication with the firmware.	Perform I/O reconfiguration. Activate reset flag in IBSPCI.XML file if required.
Unable to project %1 diagnostic registers into I/O map.	A firmware service for configuring the diagnostic registers into the I/O map has failed.	The diagnostic registers cannot be displayed in the I/O map.	Evaluate return values of the firmware services. The I/O addresses are incorrect.
%1 Unable to generate and start configuration frame.	An error occurred on loading the current configuration and starting the generated configuration frame. To locate the precise cause, the error codes of the firmware services must be evaluated.	The Interbus is not started and no I/O data are exchanged.	Depends on the error code of the firmware services.

Message	Reason	Effect	Remedy
%1 Error on starting slave ring without master ring.	An error occurred in a firmware service when starting the slave ring without starting the master. The cause depends on the return values of the firmware services.	The slave ring is not activated and no I/O data are exchanged.	Depends on the return value of the firmware services.
%1 Negative return value received from one command in the SVC file.	A negative return value was received from the firmware when executing a command from the SVC file.	The message has no effect on the execution of the SVC file. Other errors will occur, depending on which firmware service has failed.	Activate logging of the Interbus driver with level debug and check the return value of the firmware service. The physical bus configuration does not match that configured in the SVC file.

10 KUKA Service

10.1 Requesting support

Introduction

The KUKA Roboter GmbH documentation offers information on operation and provides assistance with troubleshooting. For further assistance, please contact your local KUKA subsidiary.

Information

The following information is required for processing a support request:

- Model and serial number of the robot
- Model and serial number of the controller
- Model and serial number of the linear unit (if applicable)
- Model and serial number of the energy supply system (if applicable)
- Version of the KUKA System Software
- Optional software or modifications
- Archive of the software

  For KUKA System Software V8: instead of a conventional archive, generate the special data package for fault analysis (via **KrcDiag**).
- Application used
- Any external axes used
- Description of the problem, duration and frequency of the fault

10.2 KUKA Customer Support

Availability

KUKA Customer Support is available in many countries. Please do not hesitate to contact us if you have any questions.

Argentina

Ruben Costantini S.A. (Agency)
Luis Angel Huergo 13 20
Parque Industrial
2400 San Francisco (CBA)
Argentina
Tel. +54 3564 421033
Fax +54 3564 428877
ventas@costantini-sa.com

Australia

Headland Machinery Pty. Ltd.
Victoria (Head Office & Showroom)
95 Highbury Road
Burwood
Victoria 31 25
Australia
Tel. +61 3 9244-3500
Fax +61 3 9244-3501
vic@headland.com.au
www.headland.com.au

Belgium

KUKA Automatisering + Robots N.V.
Centrum Zuid 1031
3530 Houthalen
Belgium
Tel. +32 11 516160
Fax +32 11 526794
info@kuka.be
www.kuka.be

Brazil

KUKA Roboter do Brasil Ltda.
Travessa Claudio Armando, nº 171
Bloco 5 - Galpões 51/52
Bairro Assunção
CEP 09861-7630 São Bernardo do Campo - SP
Brazil
Tel. +55 11 4942-8299
Fax +55 11 2201-7883
info@kuka-roboter.com.br
www.kuka-roboter.com.br

Chile

Robotec S.A. (Agency)
Santiago de Chile
Chile
Tel. +56 2 331-5951
Fax +56 2 331-5952
robotec@robotec.cl
www.robotec.cl

China

KUKA Robotics China Co.,Ltd.
Songjiang Industrial Zone
No. 388 Minshen Road
201612 Shanghai
China
Tel. +86 21 6787-1888
Fax +86 21 6787-1803
www.kuka-robotics.cn

Germany

KUKA Roboter GmbH
Zugspitzstr. 140
86165 Augsburg
Germany
Tel. +49 821 797-4000
Fax +49 821 797-1616
info@kuka-roboter.de
www.kuka-roboter.de

France

KUKA Automatisme + Robotique SAS
Techvallée
6, Avenue du Parc
91140 Villebon S/Yvette
France
Tel. +33 1 6931660-0
Fax +33 1 6931660-1
commercial@kuka.fr
www.kuka.fr

India

KUKA Robotics India Pvt. Ltd.
Office Number-7, German Centre,
Level 12, Building No. - 9B
DLF Cyber City Phase III
122 002 Gurgaon
Haryana
India
Tel. +91 124 4635774
Fax +91 124 4635773
info@kuka.in
www.kuka.in

Italy

KUKA Roboter Italia S.p.A.
Via Pavia 9/a - int.6
10098 Rivoli (TO)
Italy
Tel. +39 011 959-5013
Fax +39 011 959-5141
kuka@kuka.it
www.kuka.it

Japan

KUKA Robotics Japan K.K.
YBP Technical Center
134 Godo-cho, Hodogaya-ku
Yokohama, Kanagawa
240 0005
Japan
Tel. +81 45 744 7691
Fax +81 45 744 7696
info@kuka.co.jp

Canada

KUKA Robotics Canada Ltd.
6710 Maritz Drive - Unit 4
Mississauga
L5W 0A1
Ontario
Canada
Tel. +1 905 670-8600
Fax +1 905 670-8604
info@kukarobotics.com
www.kuka-robotics.com/canada

Korea

KUKA Robotics Korea Co. Ltd.
RIT Center 306, Gyeonggi Technopark
1271-11 Sa 3-dong, Sangnok-gu
Ansan City, Gyeonggi Do
426-901
Korea
Tel. +82 31 501-1451
Fax +82 31 501-1461
info@kukakorea.com

Malaysia

KUKA Robot Automation Sdn Bhd
South East Asia Regional Office
No. 24, Jalan TPP 1/10
Taman Industri Puchong
47100 Puchong
Selangor
Malaysia
Tel. +60 3 8061-0613 or -0614
Fax +60 3 8061-7386
info@kuka.com.my

Mexico

KUKA de México S. de R.L. de C.V.
Progreso #8
Col. Centro Industrial Puente de Vigas
Tlalnepantla de Baz
54020 Estado de México
Mexico
Tel. +52 55 5203-8407
Fax +52 55 5203-8148
info@kuka.com.mx
www.kuka-robotics.com/mexico

Norway

KUKA Sveiseanlegg + Roboter
Sentrumsvegen 5
2867 Hov
Norway
Tel. +47 61 18 91 30
Fax +47 61 18 62 00
info@kuka.no

Austria

KUKA Roboter Austria GmbH
Vertriebsbüro Österreich
Regensburger Strasse 9/1
4020 Linz
Austria
Tel. +43 732 784752
Fax +43 732 793880
office@kuka-roboter.at
www.kuka-roboter.at

Poland

KUKA Roboter Austria GmbH
Spółka z ograniczoną odpowiedzialnością
Oddział w Polsce
Ul. Porcelanowa 10
40-246 Katowice
Poland
Tel. +48 327 30 32 13 or -14
Fax +48 327 30 32 26
ServicePL@kuka-roboter.de

Portugal

KUKA Sistemas de Automatización S.A.
Rua do Alto da Guerra n° 50
Armazém 04
2910 011 Setúbal
Portugal
Tel. +351 265 729780
Fax +351 265 729782
kuka@mail.telepac.pt

Russia

OOO KUKA Robotics Rus
Webnaja ul. 8A
107143 Moskau
Russia
Tel. +7 495 781-31-20
Fax +7 495 781-31-19
kuka-robotics.ru

Sweden

KUKA Svetsanläggningar + Robotar AB
A. Odhners gata 15
421 30 Västra Frölunda
Sweden
Tel. +46 31 7266-200
Fax +46 31 7266-201
info@kuka.se

Switzerland

KUKA Roboter Schweiz AG
Industriestr. 9
5432 Neuenhof
Switzerland
Tel. +41 44 74490-90
Fax +41 44 74490-91
info@kuka-roboter.ch
www.kuka-roboter.ch

Spain

KUKA Robots IBÉRICA, S.A.
Pol. Industrial
Torrent de la Pastera
Carrer del Bages s/n
08800 Vilanova i la Geltrú (Barcelona)
Spain
Tel. +34 93 8142-353
Fax +34 93 8142-950
Comercial@kuka-e.com
www.kuka-e.com

South Africa

Jendamark Automation LTD (Agency)
76a York Road
North End
6000 Port Elizabeth
South Africa
Tel. +27 41 391 4700
Fax +27 41 373 3869
www.jendamark.co.za

Taiwan

KUKA Robot Automation Taiwan Co., Ltd.
No. 249 Pujong Road
Jungli City, Taoyuan County 320
Taiwan, R. O. C.
Tel. +886 3 4331988
Fax +886 3 4331948
info@kuka.com.tw
www.kuka.com.tw

Thailand

KUKA Robot Automation (M)SdnBhd
Thailand Office
c/o Maccall System Co. Ltd.
49/9-10 Soi Kingkaew 30 Kingkaew Road
Tt. Rachatheva, A. Bangpli
Samutprakarn
10540 Thailand
Tel. +66 2 7502737
Fax +66 2 6612355
atika@ji-net.com
www.kuka-roboter.de

Czech Republic

KUKA Roboter Austria GmbH
Organisation Tschechien und Slowakei
Sezemická 2757/2
193 00 Praha
Horní Počernice
Czech Republic
Tel. +420 22 62 12 27 2
Fax +420 22 62 12 27 0
support@kuka.cz

Hungary

KUKA Robotics Hungaria Kft.
Fö út 140
2335 Taksony
Hungary
Tel. +36 24 501609
Fax +36 24 477031
info@kuka-robotics.hu

USA

KUKA Robotics Corporation
51870 Shelby Parkway
Shelby Township
48315-1787
Michigan
USA
Tel. +1 866 873-5852
Fax +1 866 329-5852
info@kukarobotics.com
www.kukarobotics.com

UK

KUKA Automation + Robotics
Hereward Rise
Halesowen
B62 8AN
UK
Tel. +44 121 585-0800
Fax +44 121 585-0900
sales@kuka.co.uk

Index